Practical No: 5

A program to implement Rule Based System.

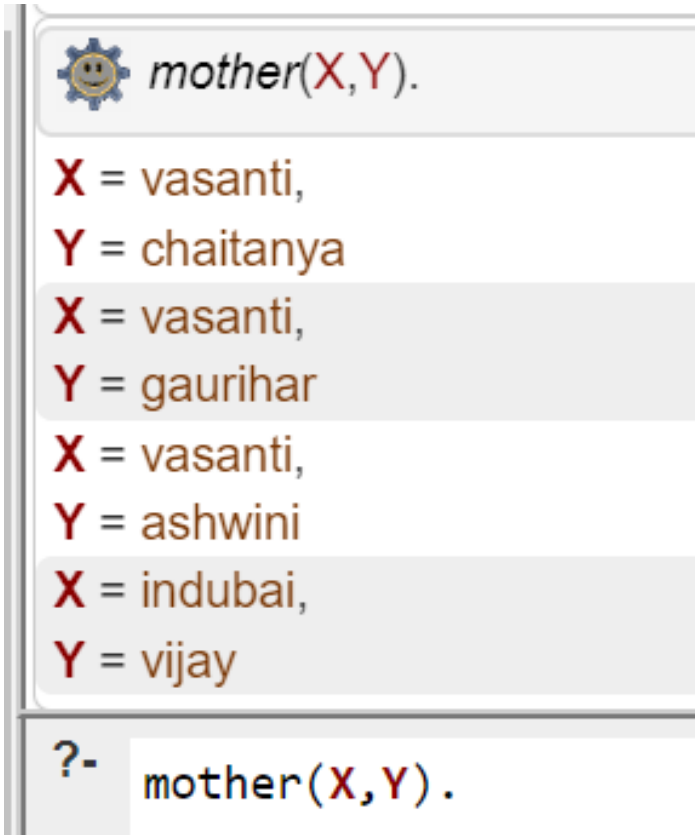
AIM: Write a program which contains three predicates: male, female, parent. Make rules for following family relations: father, mother, grandfather, grandmother, brother, sister, uncle, aunt, nephew and niece, cousin.

Code:

```
male(vijay).
male(mahadev).
male(gaurihar).
male(omkar).
male(bajrang).
male(chaitanya).

female(vasanti).
female(indubai).
female(ashwini).
female(gayatri).
female(sangita).

parent(vijay,chaitanya).
parent(vasanti,chaitanya).
parent(vijay,gaurihar).
parent(vasanti,gaurihar).
parent(vijay,ashwini).
parent(vasanti,ashwini).
parent(mahadev,vijay).
parent(indubai,vijay).

mother(X,Y):-parent(X,Y),female(X).
father(X,Y):- parent(X,Y), male(X).

grandmother(GM,X):- mother(GM,Y) ,parent(Y,X).
grandfather(GF,X):- father(GF,Y) ,parent(Y,X).

greatgrandmother(GGM,X):- mother(GGM,GM) ,parent(GM,F),parent(F,Y),parent(Y,X).
greatgrandfather(GGF,X):- father(GGF,GF) ,parent(GF,F),parent(F,Y),parent(Y,X).

sibling(X,Y):-mother(M,X), mother(M,Y),X\=Y, father(F,X), father(F,Y).
brother(X,Y):-sibling(X,Y), male(X).
sister(X,Y):-sibling(X,Y), female(X).
uncle(U,X):- parent(Y,X), brother(U,Y).
aunt(A,X):- parent(Y,X), sister(A,Y).
nephew(N,X):- sibling(S,X),parent(S,N),male(N).
niece(N,X):-sibling(S,X), parent(S,N), female(N).
cousin(X,Y):-parent(P,Y),sibling(S,P),parent(S,X).
```

OUTPUT:


father(X,Y).
X = vijay,
Y = chaitanya
X = vijay,
Y = gaurihar
X = vijay,
Y = ashwini
X = mahadev,
Y = vijay
?- father(X,Y).


mother(X,Y).
X = vasanti,
Y = chaitanya
X = vasanti,
Y = gaurihar
X = vasanti,
Y = ashwini
X = indubai,
Y = vijay
?- mother(X,Y).

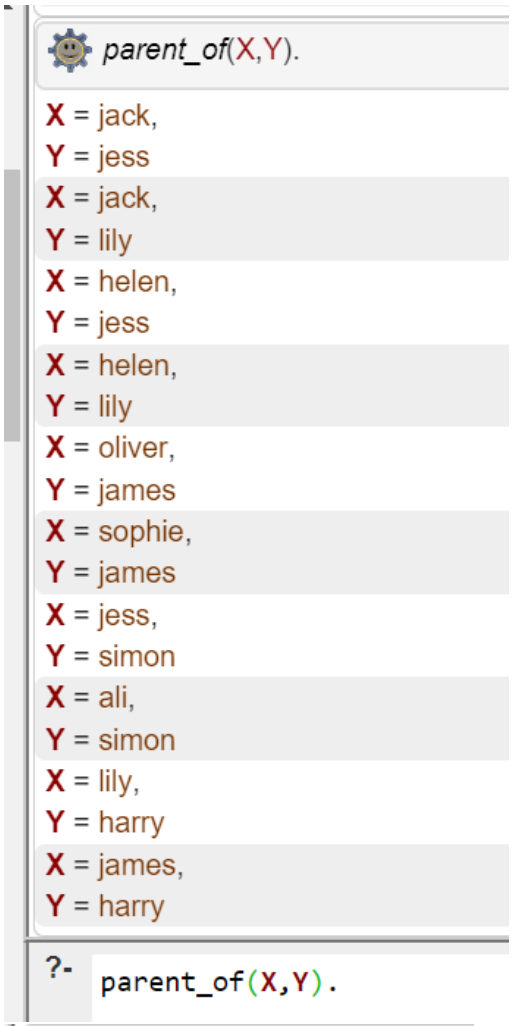

sibling(X,Y).
X = chaitanya,
Y = gaurihar
X = chaitanya,
Y = ashwini
X = gaurihar,
Y = chaitanya
X = gaurihar,
Y = ashwini
X = ashwini,
Y = chaitanya
X = ashwini,
Y = gaurihar
?- sibling(X,Y).

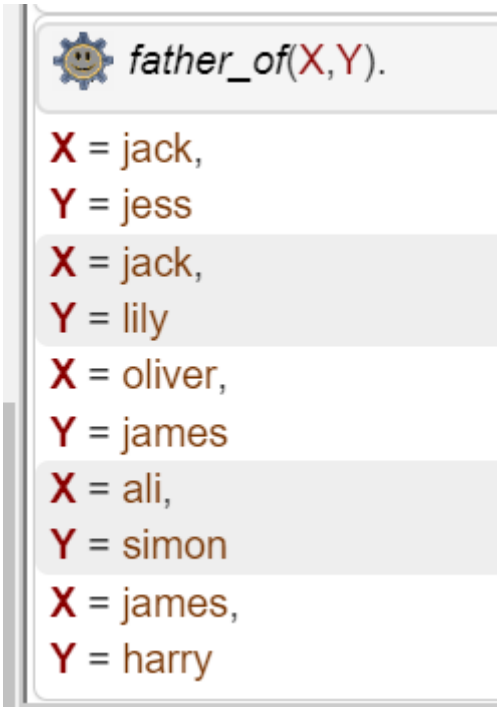
b)

Code:

```
/* Facts */
male(jack).
male(oliver).
male(ali).
male(james).
male(simon).
male(harry).
female(helen).
female(sophie).
female(jess).
female(lily).

parent_of(jack, jess).
parent_of(jack, lily).
parent_of(helen, jess).
parent_of(helen, lily).
parent_of(oliver, james).
parent_of(sophie, james).
parent_of(jess, simon).
parent_of(ali, simon).
parent_of(lily, harry).
parent_of(james, harry).

/* Rules */
father_of(X, Y):- male(X), parent_of(X, Y).
mother_of(X, Y):- female(X), parent_of(X, Y).
grandfather_of(X, Y):- male(X), parent_of(X, Z), parent_of(Z, Y).
grandmother_of(X, Y):- female(X), parent_of(X, Z), parent_of(Z, Y).
sister_of(X, Y):- female(X), father_of(F, Y), father_of(F, X), X \= Y.
sister_of(X, Y):- female(X), mother_of(M, Y), mother_of(M, X), X \= Y.
aunt_of(X, Y):- female(X), parent_of(Z, Y), sister_of(Z, X), !.
brother_of(X, Y):- male(X), father_of(F, Y), father_of(F, X), X \= Y.
brother_of(X, Y):- male(X), mother_of(M, Y), mother_of(M, X), X \= Y.
uncle_of(X, Y):- parent_of(Z, Y), brother_of(Z, X).
ancestor_of(X, Y):- parent_of(X, Y).
ancestor_of(X, Y):- parent_of(X, Z), ancestor_of(Z, Y).
```

OUTPUT:


parent_of(X,Y).
X = jack,
Y = jess
X = jack,
Y = lily
X = helen,
Y = jess
X = helen,
Y = lily
X = oliver,
Y = james
X = sophie,
Y = james
X = jess,
Y = simon
X = ali,
Y = simon
X = lily,
Y = harry
X = james,
Y = harry
?- parent_of(X,Y).


father_of(X,Y).
X = jack,
Y = jess
X = jack,
Y = lily
X = oliver,
Y = james
X = ali,
Y = simon
X = james,
Y = harry


mother_of(X,Y).
X = helen,
Y = jess
X = helen,
Y = lily
X = sophie,
Y = james
X = jess,
Y = simon
X = lily,
Y = harry