Write a program to implement Family Tree

Code

```
% FATHERS SIDE
MALE(VARUN).
MALE(KARAN).
MALE(YOGESH).
MALE(MAHENDRA).
MALE(PRASHANT).
MALE(SANJAY).
MALE(YASH).
MALE(ARVIND).

FEMALE(AMITA).
FEMALE(JYOTI).
FEMALE(DEEPAL).
FEMALE(TARUNI).
FEMALE(ARCHANA).
FEMALE(ARPITA).
FEMALE(HEMLATA).

PARENT_OF(MAHENDRA,VARUN).
PARENT_OF(AMITA,VARUN).
PARENT_OF(JYOTI,KARAN).
PARENT_OF(JYOTI,DEEPAL).
PARENT_OF(YOGESH,KARAN).
PARENT_OF(YOGESH,DEEPAL).
PARENT_OF(PRASHANT,YASH).
PARENT_OF(PRASHANT,ARPITA).
PARENT_OF(TARUNI,YASH).
PARENT_OF(TARUNI,ARPITA).
PARENT_OF(ARVIND,MAHENDRA).
PARENT_OF(ARVIND,YOGESH).
PARENT_OF(ARVIND,PRASHANT).
PARENT_OF(ARVIND,SANJAY).
PARENT_OF(HEMLATA,YOGESH).
PARENT_OF(HEMLATA,MAHENDRA).

MARRIED(YOGESH,JYOTI).
MARRIED(MAHENDRA,AMITA).
MARRIED(PRASHANT,TARUNI).
MARRIED(SANJAY,ARCHANA).
MARRIED(ARVIND,HEMLATA).
```

%SELF ROLES RULE
FATHER(X, Y) :- MALE(X), PARENT_OF(X, Y).
MOTHER(X,Y) :- FEMALE(X), PARENT_OF(X,Y).

BROTHER(X, Y) :- MALE(X), FATHER(Z, Y), FATHER(Z, X), X \= Y.
BROTHER(X, Y) :- MALE(X), MOTHER(Z, Y), MOTHER(Z, X), X \= Y.

SISTER(X, Y) :- FEMALE(X), FATHER(Z, Y), FATHER(Z, X), X \= Y.
SISTER(X, Y) :- FEMALE(X), MOTHER(Z, Y), MOTHER(Z, X), X \= Y.

UNCLE_OF(X,Y) :- PARENT_OF(Z,Y), BROTHER(Z,X).

GRANDPARENT(X, Y) :- PARENT_OF(X, Z), PARENT_OF(Z, Y).

GRANDMOTHER(X, Y) :- PARENT_OF(X, Z), PARENT_OF(Z, Y),FEMALE(X).
GRANDFATHER(X, Y) :- PARENT_OF(X, Z), PARENT_OF(Z, Y),MALE(X).

AUNT_OF(X,Y) :- FEMALE(X), FATHER(Z,Y), BROTHER(Z,W), MARRIED(W,X).

OUTPUT

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.1)

File  Edit  Settings  Run  Debug  Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% c:/users/varun/documents/prolog/prac_2_family_tree compiled 0.00 sec, -2 clauses
?-
|    father(X,varun).
X = mahendra ,

?- mother(X,varun).
X = amita ,

?- aunt_of(X,varun).
X = jyoti ;
X = jyoti ;
X = taruni ;
X = archana ;
false.

?- grandparent(X,varun).
X = arvind ;
X = hemlata ,

?- grandfather(X,varun).
X = arvind ,

?- uncle_of(X,varun).
X = yogesh ;
X = prashant ;
X = sanjay ,
```