## LAB 2 | Artificial Intelligence

**Aim: Study of RULES & UNIFICATION.**

1. Write a prolog program for the given facts and rules and answer the given question.

Code:

|  |
| --- |
| domains  patient, indication, disease=symbol  predicates  symptom(patient, indication).  hypothesis(patient, disease).  clauses  symptom("Parva",fever).  symptom("Parva",rash).  symptom("Parva",headache).  symptom("Parva",runny\_nose).  symptom("Vidhi",chills).  symptom("Vidhi",fever).  symptom("Vidhi",headache).  symptom("Vivan",runny\_nose).  symptom("Vivan",rash).  symptom("Vivan",flu).    hypothesis(Patient,measles):-symptom(Patient,fever),  symptom(Patient,cough),  symptom(Patient,conjunctivitis),  symptom(Patient,rash).    hypothesis(Patient,german\_measles):-symptom(Patient,fever),  symptom(Patient,headache),  symptom(Patient,runny\_nose),  symptom(Patient,rash).    hypothesis(Patient,flu):-symptom(Patient,fever),  symptom(Patient, headache),  symptom(Patient,body\_ache),  symptom(Patient,chills).    hypothesis(Patient,common\_cold):-symptom(Patient,headache),  symptom(Patient,sneezing),  symptom(Patient,sore\_throat),  symptom(Patient,chills),  symptom(Patient,runny\_nose).    hypothesis(Patient,mumps):-symptom(Patient,fever),  symptom(Patient,swollen\_glands).    hypothesis(Patient,chicken\_pox):-symptom(Patient,fever),  symptom(Patient,rash),  symptom(Patient,body\_ache),  symptom(Patient,chills). |

Question : Identify patients with any particular disease based on rules and facts given above.



1. Write a program for a family tree given question which contains three predicates: male, female, parent.

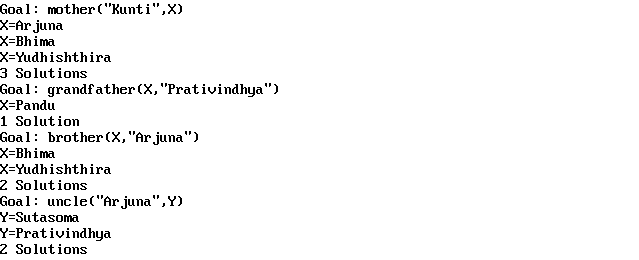
Make rules for family relations : father , mother, grandfather, grandmother, brother, sister, uncle, aunt, nephew and niece.

Code:

|  |
| --- |
| predicates  male(symbol).  female(symbol).  parent(symbol,symbol).  father(symbol,symbol).  mother(symbol,symbol).  wife(symbol,symbol).  grandfather(symbol,symbol).  grandmother(symbol,symbol).  brother(symbol,symbol).  sister(symbol,symbol).  uncle(symbol,symbol).  aunt(symbol,symbol).  nephew(symbol,symbol).  niece(symbol,symbol).    clauses  male("Pandu").  male("Nakula").  male("Sahadeva").  male("Arjuna").  male("Bhima").  male("Yudhishthira").  male("Satanika").  male("Shrutasena").  male("Shrutakarma").  male("Abhimanyu").  male("Iravan").  male("Babruvahana").  male("Sutasoma").  male("Prativindhya").    female("Madri").  female("Kunti").  female("Draupadi").  female("Subhadra").  female("Ulupi").  female("Chitrangada").    parent("Pandu","Nakula").  parent("Pandu","Sahadeva").  parent("Pandu","Arjuna").  parent("Pandu","Bhima").  parent("Pandu","Yudhishthira").  parent("Madri","Nakula").  parent("Madri","Sahadeva").  parent("Kunti","Arjuna").  parent("Kunti","Bhima").  parent("Kunti","Yudhishthira").  parent("Nakula","Satanika").  parent("Draupadi","Satanika").  parent("Sahadeva","Shrutasena").  parent("Draupadi","Shrutasena").  parent("Arjuna","Shrutakarma").  parent("Arjuna","Abhimanyu").  parent("Arjuna","Iravan").  parent("Arjuna","Babruvahana").  parent("Draupadi","Shrutakarma").  parent("Subhadra","Abhimanyu").  parent("Ulupi","Iravan").  parent("Chitrangada","Babruvahana").  parent("Bhima","Sutasoma").  parent("Draupadi","Sutasoma").  parent("Yudhishthira","Prativindhya").  parent("Draupadi","Prativindhya").    father(X,Y):-parent(X,Y),male(X).  mother(X,Y):-parent(X,Y),female(X).  wife(X,Y):-parent(X,Z),parent(Y,Z),  male(X),female(Y).  grandfather(X,Y):-father(X,Z),father(Z,Y).  grandmother(X,Y):-mother(X,Z),father(Z,Y).  brother(X,Y):-father(A,X),father(A,Y),  mother(B,X),mother(B,Y),  male(X),not(X=Y).  sister(X,Y):-father(A,X),father(A,Y),  mother(B,X),mother(B,Y),  female(X),not(X=Y).  uncle(X,Y):-father(Z,Y),brother(X,Z).  aunt(X,Y):-father(Z,Y),brother(B,Z),wife(B,X).  nephew(X,Y):-father(Z,Y),brother(X,Z),  male(X),male(Y).  niece(X,Y):-father(Z,Y),brother(X,Z),  male(X),female(Y). |

Output :







1. Write a prolog program for the given facts and rules, trace the given goals.

Code:

|  |
| --- |
| domains  course, level, material, component, person = symbol  predicates  is(course,level).  available(course,material).  has(course,component).  takes(person,course).  hypothesis(person,course).  clauses  is("hardware","easy").  is("logic","not easy").  is("graphics","easy").    has("graphics","8 credits").  has("graphics","lab component").    available("hardware","Books").  available("database","Books").    takes("Mary","compilers").    hypothesis(X,Y):-takes(X,Y),is(Y,"easy"),available(Y,"Books").  hypothesis(X,Y):-takes(X,Y),has(Y,"8 credits"),has(Y,"lab component"). |

Goals:

1. Does Mary take a graphics course?

I/p & O/p:



1. Which course Mary takes?

I/p & O/p:



1. Who takes graphics course?

I/p & O/p:

