








## PROGRAM 1: PROLOG PROGRAM FOR COLLEGE KNOWLEDGE

```
class(fy,it).
class(sy,it).
class(ty,it).
college(rscoe).
city(rscoe,pune).
state(rscoe,mh).
branch(it,rscoe).
branch(cs,rscoe).
branch(etc,rscoe).
branch(civil,rscoe).
branch(mech,rscoe).
subject(fy,cpp).
subject(sy, ds).
subject(ty, al).
location(X,Y):-city(C,X), state(C,Y).
learns(X,Y):- class(C,X), subject(C,Y).
```

 <i>branch(Which,rscoe).</i>
<b>Which</b> = it
<b>Which</b> = cs
<b>Which</b> = etc
<b>Which</b> = civil
<b>Which</b> = mech
 <i>location(X,mh)</i>
<b>X</b> = pune
 <i>subject(X, al).</i>
<b>X</b> = ty
 <i>subject(X, ds).</i>
<b>X</b> = sy
 <i>city(rscoe,Where).</i>
<b>Where</b> = pune
 <i>state(rscoe,Where).</i>
<b>Where</b> = mh
 <i>class(X,it).</i>
<b>X</b> = fy
<b>X</b> = sy
<b>X</b> = ty

## PROGRAM 2: PROLOG PROGRAM FOR RELATIONS KNOWLEDGE

*parent(x,y).*

*parent(z,x).*

*child(X,Y):-parent(Y,X).*

*grandparent(Z,Y):-parent(Z,X),parent(X,Y).*

*friend(p,y).*





*friend(X,Y):-friend(Y,X).*

*likes(p,sing).*

*likes(y,cricket).*

*classmates(p,y).*

*classmates(X,Y):-classmates(Y,X).*

 <i>likes(y,What).</i>
<b>What</b> = cricket
 <i>child(y,Of).</i>
<b>Of</b> = x
 <i>parent(x,Child).</i>
<b>Child</b> = y
 <i>grandparent(z,GrandChild).</i>
<b>GrandChild</b> = y
 <i>classmates(p,y).</i>
true
 <i>classmates(y,p).</i>
true
 <i>classmates(y,Who).</i>
<b>Who</b> = p

### PROGRAM 3: PROLOG PROGRAM FOR TEACHER STUDENT KNOWLEDGE

*studies(charlie, csc135).*

*studies(olivia, csc135).*

*studies(jack, csc131).*

*studies(arthur, csc134).*

*teaches(kirke, csc135).*

*teaches(collins, csc131).*

*teaches(collins, csc171).*

*teaches(juniper, csc134).*

*professor(X, Y) :- teaches(X, C), studies(Y, C).*

 <code>studies(jack,X).</code>
<code>X = csc131</code>
 <code>studies(X,csc134).</code>
<code>X = arthur</code>
 <code>teacher(collins,What).</code>
<code>What = csc131</code>
<code>What = csc171</code>
 <code>teacher(Who,csc135).</code>
<code>Who = kirke</code>
 <code>professor(kirke,olivia).</code>
<code>true</code>
 <code>professor(Who,arthur).</code>
<code>Who = juniper</code>
 <code>professor(collins,Who).</code>
<code>Who = jack</code>





#### PROGRAM 4: PROLOG PROGRAM FOR MIN MAX

*find\_max*(X,Y,X):-X>Y, !.

*find\_max*(X,Y,Y):-Y>X.

*find\_min*(X,Y,X):-X<Y,!.

*find\_min*(X,Y,Y):-Y<X.

 <code>find_max(100,200,X).</code>
<code>X = 200</code>
 <code>find_min(100,200,X).</code>
<code>X = 100</code>
 <code>find_max(400,200,X).</code>
<code>X = 400</code>
 <code>find_min(400,200,X).</code>
<code>X = 200</code>

## PROGRAM 5: PROLOG PROGRAM FOR BIKES

*bike(ktm).*

*bike(bike1).*

*bike(bike2).*

*bike(bike3).*

*location(bike1,city1).*

*location(bike1,city2).*

*location(bike2,city2).*

*location(bike3,city3).*

*category(bike1,electric).*

*category(bike2,petrol).*

*category(bike3,pertol).*

*price(bike1,80000).*

*price(bike2,70000).*


*price(bike3,60000).*

*find\_max(A,B,A):-price(A,X),price(B,Y),X>=Y, ! .*


*find\_max(A,B,B):-price(A,X),price(B,Y),Y>X.*

*find\_min(A,B,A):-price(A,X),price(B,Y),X<Y, ! .*


*find\_min(A,B,B):-price(A,X),price(B,Y),Y<X.*

 *find\_max*(bike1,bike2,What).

**What** = bike1

 *find\_max*(bike2,bike3,What).


**What** = bike2

 *find\_max*(bike1,bike3,What).

**What** = bike1

 *find\_min*(bike1,bike2,What).


**What** = bike2

 *find\_min*(bike2,bike3,What).

**What** = bike3

 *find\_min*(bike1,bike3,What).

**What** = bike3


 *location*(bike1,X).

**X** = city1

**X** = city2

 *location*(bike2,X).

**X** = city2

 *location*(bike3,X).

**X** = city3