1. Family Tree

Code:

```
parent(john, mary).
parent(john, tom).
parent(mary, alice).
parent(tom, bob).
male(john).
male(tom).
male(bob).
female(mary).
female(alice).

father(X, Y) :- parent(X, Y), male(X).
mother(X, Y) :- parent(X, Y), female(X).
sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.
```

Query:

```
?- father(john, mary).
```

2. Concatenate Lists

Code:

```
conc([], L, L).
conc([H|T], L2, [H|R]) :- conc(T, L2, R).
```

Query:

```
?- conc([1,2], [3,4], X).
```

3. Reverse List

Code:

```
reverse([], []).
reverse([H|T], R) :- reverse(T, RevT), conc(RevT, [H], R).
conc([], L, L).
conc([H|T], L2, [H|R]) :- conc(T, L2, R).
```

Query:

```
?- reverse([1,2,3], R).
```

4. Sum of Two Numbers

Code:

```
sum(X, Y, S) :- S is X + Y.
```

Query:

```
?- sum(5, 3, S).
```

5. Maximum of Two Numbers

Code:

```
max(X, Y, X) :- X >= Y.
max(X, Y, Y) :- Y > X.
```

Query:
?- max(7, 4, M).

6. Factorial

Code:
```
factorial(0, 1).
factorial(N, F) :- N > 0, N1 is N - 1, factorial(N1, F1), F is N * F1.
```

Query:
?- factorial(5, F).

7. Fibonacci Series

Code:
```
generate_fib(0, 0).
generate_fib(1, 1).
generate_fib(N, T) :- N > 1, N1 is N - 1, N2 is N - 2,
                generate_fib(N1, T1), generate_fib(N2, T2), T is T1 + T2.
```

Query:
?- generate_fib(6, T).

8. Power Function

Code:
```
power(_, 0, 1).
power(Num, Pow, Ans) :- Pow > 0, P1 is Pow - 1, power(Num, P1, A1), Ans is Num * A1.
```

Query:
?- power(2, 3, Ans).

9. Multiply Two Numbers

Code:
```
multi(N1, N2, R) :- R is N1 * N2.
```

Query:
?- multi(4, 5, R).

10. Membership Check

Code:
```
memb(X, [X|_]).
memb(X, [_|T]) :- memb(X, T).
```

Query:
?- memb(3, [1,2,3,4]).

11. Sum of List

Code:
```
sumlist([], 0).
sumlist([H|T], S) :- sumlist(T, ST), S is H + ST.
```

Query:
?- sumlist([1,2,3], S).

12. Even/Odd Length List

Code:
```
evenlength([]).
evenlength([_,_|T]) :- evenlength(T).

oddlength([_]).
oddlength([_,_|T]) :- oddlength(T).
```

Query:
?- evenlength([1,2,3,4]).

13. Maximum in List

Code:
```
maxlist([X], X).
maxlist([H|T], M) :- maxlist(T, TM), M is max(H, TM).
```

Query:
?- maxlist([3, 5, 2, 9], M).

14. Insert at Nth Position

Code:
```
insert(0, X, L, [X|L]).
insert(I, X, [H|T], [H|R]) :- I > 0, I1 is I - 1, insert(I1, X, T, R).
```

Query:
?- insert(2, x, [a,b,c,d], R).

15. Delete at Nth Position

Code:
```
delete(0, [_|T], T).
delete(N, [H|T], [H|R]) :- N > 0, N1 is N - 1, delete(N1, T, R).
```

Query:
?- delete(2, [a,b,c,d], R).