Q1.
```prolog
sum(X,Y):-
    S is X+Y,
    write(S).
```

Q2.
```prolog
fact(0,1).
fact(N,F):-
(
 N>0 ->
 (
  N1 is N-1,
  fact(N1,F1),
  F is N*F1
 )
 ;

 N<0 ->
 (
  N1 is N+1,
  fact(N1,F1),
  F is N*F1
 )
).
```

Q3.
```prolog
fib(0, 1) :- !.

fib(1, 1) :- !.

fib(N, F) :-

        N > 1,

        N1 is N-1,

        N2 is N-2,

        fib(N1, F1),

        fib(N2, F2),

        F is F1+F2.
```

Q4.
```prolog
max(X,Y):-
(
 X=Y ->
  write('both are equal')
 ;
 X>Y ->
  (
  Z is X,
  write(Z)
  )
  ;
  (
  Z is Y,
```

```prolog
    write(Z)
  )
).
```

Q5.
```prolog
 mem(X,[X|_]).
 mem(X,[_|T]):- mem(X,T).
 insert(L,[X|Y],[L|_]).
 insert(L,P,[X|Y],[X|M]):-
 P>1,
 P1 is P-1,
 insert(L,P1,Y,M).
 insert(L,1,[X|Y],M):- append([L],[X|Y],M).
```

Q6.
```prolog
delte(1,[_|T],T).
    delte(P,[X|Y],[X|R]):-
    P1 is P-1,
    delte(P1,Y,R).
```

Q8.
```prolog
conc([],L,L).
conc([X|M],N,[X|Q]):-
 conc(M,N,Q).palind([]):- write('palindrome').
palind([_]):- write('palindrome').
palind(L) :-
 append([H|T], [H], L),
 palind(T)
 ;
 write('Not a palindrome').
```

Q9.
```prolog
palind([]):- write('palindrome').
palind([_]):- write('palindrome').
palind(L) :-
 append([H|T], [H], L),
 palind(T)
 ;
 write('Not a palindrome').max(X,Y,R):-
 X>=Y ->
  R is X,
  write(R)
 ;
  R is Y,
  write(R).
```

```prolog
/* with list. */
grandiose([H|T],R):-
 H>T ->
  R is H,
  write(R)
```

```prolog
  ;
  R is T,
  write(T).
```

Q10.

```prolog
max(X,Y,R):-
 X>=Y ->
  R is X,
  write(R)
 ;
  R is Y,
  write(R).
max2([H],H).
max2([H|T],R):-
 max2(T,M1),
 H>=M1,
 R is H,!.
max2([H|T],R):-
 max2(T,M1),
 H<M1,
 R is M1.
grandiose([H|T],R):-
 H>T ->
  R is H,
  write(R)
 ;
  R is T,
  write(T).
```

Q11.

```prolog
max2([H],H).
max2([H|T],R):-
 max2(T,M1),
 H>=M1,
 R is H,!.
max2([H|T],R):-
 max2(T,M1),
 H<M1,
 R is M1.
```