```
%Q1
parent(john,estelle).
parent(john, rob).
parent(peter, grace).
parent(peter,george).
parent(george, john).
parent(george, mary).
parent(george, jay).
parent(sue, rob).
parent(sue, estelle).
parent(ida, george).
parent(ida, grace).
parent(estelle, john).
parent(estelle, mary).
parent(estelle, jay).
male(john).
male(peter).
male(rob).
male(george).
male(jay).
male(john).
female(sue).
female(ida).
female(estelle).
female(grace).
female(mary).

grandparent(Grandparent,Grandchild):-
  parent(Grandparent, Z),
     parent(Z, Grandchild).

grandfather(Grandfather, Grandchild):-
   male(Grandfather),
   parent(Grandfather, Z),
   parent(Z, Grandchild).

uncle(Uncle, Nephew):-
   grandparent(Y,Nephew),
   parent(Y, Uncle),
   \+ parent(Uncle, Y),
   male(Uncle).

mother_in_law(Mother_in_law, Child_in_law):-
   parent(Child_in_law, Y),
   grandparent(Mother_in_law, Y),
   \+ parent(Mother_in_law, Child_in_law),
   female(Mother_in_law).

brother(brother, bro):-
   \+ brother = bro,
   parent(Y, brother),
   parent(Y, bro),
   male(brother).

two_brother(brother1,brother2, bro):-
   male(brother1),
   male(brother2),
   \+ brother1 = brother2,
   \+ brother1 = bro,
   \+ brother2 = bro,
   parent(Y, brother1),
   parent(Y, brother2),
   parent(Y, bro).
```

%Q2

%Q3

```
last(X,[Xs]):-
        X = Xs.
    last(X, [_|Tail]):-
            last(X, Tail).
```

%Q4

```
fib(1, 0) :- !.
fib(2, 1) :- !.
fib(N, F) :-
    N>2,
    N1 is N-1,
    N2 is N-2,
    fib(N1, F1),
    fib(N2, F2),
    F is F1+F2.
```

%Q5

```
sum([],0).
sum([Head|Subset], Sum):-
    sum(Subset, Subsum),
    Sum is Head+Subsum.

count([],0).
count([_|Tail],Count):-
    count(Tail, Subcount),
    Count is Subcount+1.

mean(List, Mean):-
    sum(List, S),
    count(List, C),
    Mean is S/C.


min([X], X) :- !.
min([X,Y|Tail], N):-
    ( X > Y ->
        min([Y|Tail], N);
        min([X|Tail], N)
    ).

max([X], X) :- !.
max([X,Y|Tail], N):-
    ( X < Y ->
        max([Y|Tail], N);
        max([X|Tail], N)
    ).
```