1. (8 pts) Given the following clauses, list all answers returned by the queries. Note, these facts may not be factually accurate.

kardashian(kim).
kardashian(kourtney).
boy(scott).
married(X,Y):- X = kanye, kardashian(Y).
married(X,Y):- boy(X), kardashian(Y), !.
married(kris,kim).
child(penelope).
child(north).
sibling(X,Y):- child(X), child(Y).
family(X,Y):- married(X,Y), married(Y, X).
family(X,Y):- kardashian(X), kardashian(Y), X \neq Y.

a. (3 pts) ?- married(X,Y).
b. (1 pt) ?- \+child(kim).
c. (2 pts) ?- family(X,Y).
d. (2 pts) ?- sibling(X,Y).
Unification:
(2 pts) Show the variables bindings (values assigned) if the following queries succeed. Use false otherwise.
1. \( f(X, a, h, g(Y)) = f(h(Z), Z, h(W)) \).
2. \( p(X, g(b, Y), L) = p(a, g(Z, f(X)), [X|Zs]) \).

Prolog Programming:
(10 pts) Implement a prolog predicate \( \text{takeNth}(N, L, M) \) such that \( \text{takeNth}(N, L, M) \) is true if and only if \( M \) is the list obtained by taking every \( N \)th element from the list \( L \). For example the query:

\[\text{?- takeNth}(2, [1,2,3,4,5,6,7,8], M).\]
\( M = [1, 3, 5, 7] \).

\[\text{?- takeNth}(3, [1,2,3,4,5,6,7,8], M).\]
\( M = [1, 4, 7] \).

*Hint: Consider writing a helper predicate that drops the first \( N \) elements of the list.*