Prolog Diners Problem

Mr and Mrs Astor, Mr and Mrs Blake, Mr and Mrs Crane, and Mr and Mrs Davis were seated around a circular table. Mrs Astor was insulted by Mr Blake, who sat next to her on her left. Mr Blake was insulted by Mrs Crane, who sat opposite him across the center of the table. Mrs Crane was insulted by the hostess, who was the only person to sit next to each one of a married couple. The hostess was insulted by the only person to sit next to each one of two men. Who insulted the hostess? Mrs. Davis is the hostess and she is seated at place 0.

Assume the table is numbered as in the diagram. It might help to use the following convention for naming variables:
Mr. Astor - A
Mrs. Astor - MA
Mr. Blake - B
Mrs. Blake - MB
Mr. Crane - C
Mrs. Crane - MC
Mr. Davis - D
Mrs. Davis - MD
Prolog Diners Problem – Partial Solution 1

places([place(_,0), place(_,1), place(_,2), place(_,3), place(_,4), place(_,5), place(_,6), place(_,7)]).

dinner(Y) :- places(Places),
    member(place(mrAstor, A), Places),
    member(place(mrsAstor, MA), Places),
    member(place(mrBlake, B), Places),
    member(place(mrsBlake, MB), Places),
    member(place(mrCrane, C), Places),
    member(place(mrsCrane, MC), Places),
    member(place(mrDavis, D), Places),
    member(place(mrsDavis, MD), Places),
    print_places(Places).

print_places([]).
print_places([A | B]) :- write(A), nl, print_places(B).
Prolog Diners Problem – Partial Solution 2

places([place(7,0), place(7,1), place(7,2), place(7,3), place(7,4), place(7,5), place(7,6), place(7,7)]).

dinner(Y) :- places(Places),
    member(place(mrAstor, A), Places),
    member(place(mrsAstor, MA), Places),
    member(place(mrBlake, B), Places), B is ((MA + 1) mod 8),
    member(place(mrsBlake, MB), Places),
    member(place(mrCrane, C), Places),
    member(place(mrsCrane, MC), Places), MC is ((B + 4) mod 8),
    member(place(mrDavis, D), Places),
    member(place(mrsDavis, MD), Places), MD is 0,
    print_places(Places).

print_places([]).
print_places([A | B]) :- write(A), nl, print_places(B).
Prolog Diners Problem – Partial Solution 3

places([place(_,0), place(_,1), place(_,2), place(_,3), place(_,4), place(_,5), place(_,6), place(_,7)]).

married(mrAstor, mrsAstor).
married(mrBlake, mrsBlake).
marrried(mrCrane, mrsCrane).
marrried(mrDavis, mrsDavis).
marrried(mrsAstor, mrAstor).
marrried(mrsBlake, mrBlake).
marrried(mrsCrane, mrCrane).
marrried(mrsDavis, mrDavis).

member(place(I, 0), Places),
member(place(J, 1), Places),
member(place(K, 2), Places),
member(place(L, 3), Places),
member(place(M, 4), Places),
member(place(N, 5), Places),
member(place(O, 6), Places),
member(place(P, 7), Places),
notted(married(I, K)),
notted(married(J, L)),
notted(married(K, M)),
notted(married(L, N)),
notted(married(M, O)),
notted(married(N, P)),
notted(married(O, I)),
print_places(Places).

print_places([ ]).
print_places([A | B]) :- write(A), nl, print_places(B).
Prolog Diners Problem – Final Solution

places([place(_,0), place(_,1), place(_,2), place(_,3), place(_,4), place(_,6), place(_,7)]).

married(mrAstor, mrsAstor).
mixed(mrBlake, mrsBlake).
mixed(mrCrane, mrsCrane).
mixed(mrDavis, mrsDavis).
mixed(mrsAstor, mrAstor).
mixed(mrsBlake, mrBlake).
mixed(mrsCrane, mrCrane).
mixed(mrsDavis, mrDavis).
male(mrAstor).
male(mrBlake).
male(mrCrane).
male(mrDavis).

dinner(Y) :- places(Places),
    member(place(mrAstor, A), Places),
    member(place(mrsAstor, MA), Places),
    member(place(mrBlake, B), Places), B is ((MA + 1) mod 8),
    member(place(mrsBlake, MB), Places),
    member(place(mrCrane, C), Places),
    member(place(mrsCrane, MC), Places), MC is ((B + 4) mod 8),
    member(place(mrDavis, D), Places),
    member(place(mrsDavis, MD), Places), MD is 0,
    member(place(I, 0), Places),
    member(place(J, 1), Places),
    member(place(K, 2), Places),
    member(place(L, 3), Places),
    member(place(M, 4), Places),
    member(place(N, 5), Places),
    member(place(O, 6), Places),
    member(place(P, 7), Places),
    married(P, J),
    not(married(I, K)),
    not(married(J, L)),
    not(married(K, M)),
    not(married(L, N)),
    not(married(M, O)),
    not(married(N, P)),
    not(married(O, I)),
    member(place(Y, Z), Places),
    member(place(Zq, Q), Places), Q is (Z+1) mod 8,
    member(place(Zr, R), Places), R is (Z+2) mod 8,
    member(place(Zs, S), Places), S is (Z+3) mod 8,
    member(place(Zt, T), Places), T is (Z+4) mod 8,
    member(place(Zu, U), Places), U is (Z+5) mod 8,
    member(place(Zv, V), Places), V is (Z+6) mod 8,
    member(place(Zw, W), Places), W is (Z+7) mod 8,
    betweenMen(Zq, Q),
    not(betweenMen(Zq, Zs)),
    not(betweenMen(Zr, Zt)),
    not(betweenMen(Zs, Zu)),
    not(betweenMen(Zt, Zv)),
    not(betweenMen(Zu, Zw)),
    not(betweenMen(Zv, Y)),
    not(betweenMen(Y, Zr)),
    print_places(Places).