```
% Author:
% Date: 26/03/2015
straight_chain_alkane(1,[carb(h,h,h,h)]).
straight_chain_alkane(2,[carb(h,h,h,c),carb(c,h,h,h)]).
straight_chain_alkane(N,[H,carb(c,h,h,c)|T]):-
                        N>2,
                        N2 is N-1,
                        straight_chain_alkane(N2,[H|T]).

branched_alkane(4,[carb(h,h,h,c),carb(c,c1h3,h,c),carb(c,h,h,h)]).
branched_alkane(N,BA):-
            N>4,
            N1 is N - 3,
            mc_generator(N1,1,Mid ,Com ),
            mid_chain(Mid ,A),
            N2 is Mid+2,
            sums(Com ,Com1),
            constructer(A,Com1,B),
            valid_check(B,N2),
            append([carb(h,h,h,c)],B,BB),
            append(BB,[carb(c,h,h,h)],BA).

constructer(Mid ,Nums,Ans):-
              findall(A,num_seperate(Mid , Nums,A),A2),
              dup_remover(A2,A3),
              break_list(A3,Ans).

dup_remover([],[]).
dup_remover([Elm|T],A):-
              member(Elm,T),
              dup_remover(T,A),
```

```
                !.
dup_remover([Elm|T],Ans):-
                \+ member(Elm,T),
                dup_remover(T,A),
                Ans = [Elm|A],
                !.
num_seperate([carb(c,h,h,c)],[1,1],[carb(c,c1h3,c1h3,c)]).
num_seperate(ML,[T],Ans):-
                    carb_attacher(ML,T,Ans).
num_seperate(Mid,[H|T],A):-
                    T\=[],
                    carb_attacher(Mid,H,Out),
                    num_seperate(Out,T,A).

carb_attacher(H,0,H).
carb_attacher([H],X,A):-
                add_branch_to_carbon(H,X,A).
carb_attacher([H|T],X,[H|Z]):-
                    carb_attacher(T,X,L),
                    flatten(L,Z).
carb_attacher([H|T],X,Sol2):-
                    add_branch_to_carbon(H,X,M),
                    \+ member(M,T),
                    T\==[],
                    Sol = [M|T],
                    flatten(Sol,Sol2).

mid_chain(1,[carb(c,h,h,c)]).

mid_chain(N,[carb(c,h,h,c)|T]):-
                N > 1,
```

```
                N2 is N - 1,
                mid_chain(N2 ,T).

valid_check(L,MN):-
            valid(L,MN,1),
            reverse(L,L1),
            valid(L1,MN,1).

add_branch_to_carbon(A,0,A).
add_branch_to_carbon(carb(c,h,h,c),N,carb(c,A,h,c)):-
                        branch_name(N,A).
add_branch_to_carbon(carb(c,NH,h,c),N,carb(c,NH,A,c)):-
                            NH \= h,
                            branch_name(N,A),
                            N2 is N-1,
                            (add_branch_to_carbon(carb(c,NH,h,c),N2,carb(c,NH,A2,c)),A2
\= NH).

mc_generator(N,Acc,N,Acc):-
            N1 is (N+1)*2,
             Acc=<N1.
            % Acc=<N*2.


mc_generator(N,Acc,N1,N2):-
              NC is (N+1)*2,
              Acc=<NC,
            %   Acc=<N*2,
              NA is N -1,
              NB is Acc +1,
              mc_generator(NA,NB,N1,N2).
```

```
breakdown(0,[]).
breakdown(N,[H|T]) :-
                range(1,N,H),
                M is N - H,
                breakdown(M,T).

range(Low,High,_) :-
   Low > High,
   !,
   fail.
range(Low,_,Low).
range(Low,High,Out) :-
   Current is Low + 1,
   range(Current,High,Out).

branch_name(0,h).
branch_name(S,N):-
              S >0,
              HS is S*2+1,
              atomic_list_concat([c,S,h,HS],N).


extractor(S,N):-
           sub_atom(S, 1, 1, 2, A),
           atom_number(A,N).

valid([],_,_).
valid([carb(c,h,h,c)|T],MN,N):-
                        N1 is N+1,
                        valid(T,MN,N1).
```

```
valid([carb(c,X,_,c)|T],MN,N):-
                        X\==h,
                        extractor(X,N1),
                        N2 is N+1,
                        N3 is N1 +N2,
                        MN>=N3,
                        valid(T,MN,N2).



sums(A,B):-
        findall(X,breakdown(A,X),C),
        sums_help(C,[],Y),
        break_list(Y,B).
sums_help([],Y,Y).
sums_help([H|T],Y,B):-
                permutation(H,X),
                member(X,T),
                sums_help(T,Y,B),
                !.
sums_help([H|T],Y,B):-
                permutation(H,X),
                \+member(X,T),
                sums_help(T,[H|Y],B),
                !.


break_list([],_):=
               fail.
break_list([H|_],H).
break_list([_|T],Ans):-
                   break_list(T,Ans).
```

```
isomers(N,A):-
        N<4,
        straight_chain_alkane(N,A)
        .
isomers(N,A):-
        N>3,
        findall(B,branched_alkane(N,B),C),
        remove_ht(C,[],L),
        isomer_help(L,[],X),
        straight_chain_alkane(N,X2),
        A=[X2|X].

remove_ht([],A,A).
remove_ht([[_|T]|T2],Acc,Out):-
            last(T,Last),
            delete(T,Last,NL),
            remove_ht(T2,[NL|Acc],Out).

isomer_help([],A,A).
isomer_help([H|T],A,B):-
            (\+member(H,T),
            reverse(H,H1),
            \+member(H1,T)),
            append([carb(h,h,h,c)],H1,X),
            append(X,[carb(c,h,h,h)],Y),
            isomer_help(T,[Y|A],B).
isomer_help([H|T],A,B):-
            (member(H,T);
            reverse(H,H1),
            member(H1,T)),
            isomer_help(T,A,B).
```