solve(Names, Hairs, Countries, Clues, Solution) :-
  guess a solution
  permutation(Hairs, HairsSol),
  permutation(Countries, CountriesSol),
represent solution as a list of triples
  triples(Names, HairsSol, CountriesSol, Solution),
represent clues as triples
cue(sweden, blond) becomes triple(_, blond, sweden)
  process(Clues, Names, Hairs, Countries, ClueTriples),
test that the solution satisfies all triples
  subset(ClueTriples, Solution).

solve2(Names, Hairs, Countries, Clues, Solution) :-
  process(Clues, Names, Hairs, Countries, ClueTriples),
represent solution as a list of triples with variables
  triples(Names, HairsSol, CountriesSol, Solution),
fill in the clues in the solution
  subset(ClueTriples, Solution),
guess only the remaining holes
  permutation(Hairs, HairsSol),
  permutation(Countries, CountriesSol).
After subset there are two solutions:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>pamela</td>
<td>black</td>
<td>X</td>
</tr>
<tr>
<td>karin</td>
<td>brown</td>
<td>sweden</td>
</tr>
<tr>
<td>igor</td>
<td>Y</td>
<td>russia</td>
</tr>
<tr>
<td>otto</td>
<td>Z</td>
<td>germany</td>
</tr>
<tr>
<td>erik</td>
<td>blue</td>
<td>finland</td>
</tr>
</tbody>
</table>

Permutation only fills in the blanks (variables).
Turning three lists into a list of triples:

\[
\text{triples}([], [], [], []). \\
\text{triples}([N|Ns], [H|Hs], [C|Cs], [\text{triple}(N,H,C)|Ts]) :- \\
\hspace{1em}\text{triples}(Ns, Hs, Cs, Ts).
\]

Processing the clues into clue-triples one by one:

\[
\text{process}([], _, _, _, []). \\
\text{process}([C|Cs], Names, Hairs, Countries, [CT|CTs]) :- \\
\hspace{1em}\text{proc}(C, Names, Hairs, Countries, CT), \\
\hspace{1em}\text{process}(Cs, Names, Hairs, Countries, CTs).
\]

Processing a clue:

\[
\text{proc}(\text{clue}(A,B), Names, Hairs, Countries, \text{triple}(X,Y,Z)) :- \\
\hspace{1em}( \text{member}(A, Names), X=A \\
\hspace{1em}\quad; \text{member}(A, Hairs), Y=A \\
\hspace{1em}\quad; \text{member}(A, Countries), Z=A \\
\hspace{1em}), \\
\hspace{1em}( \text{member}(B, Names), X=B \\
\hspace{1em}\quad; \text{member}(B, Hairs), Y=B \\
\hspace{1em}\quad; \text{member}(B, Countries), Z=B \\
\hspace{1em}).
\]

One of \(X\), \(Y\) and \(Z\) remains a variable.
Check that the solution satisfies all clues.
For one clue, this check can be done using member,
therefore the predicate that checks all clues is named subset.

\[
\text{subset([], _).}
\]
\[
\text{subset([Clue|Clues], Solution) :-}
\]
\[
\quad \text{member(Clue, Solution),}
\]
\[
\quad \text{subset(Clues, Solution).}
\]

The set-library predicate \text{subset/2} (SICStus) does not work well here.