```
/* Michael E Sparks, 10-16-20

   A brief implementation of ELIZA, demonstrating basic
   pattern matching in Prolog.
   w() ~ single-word item
   s() ~ segment item
*/

eliza(Action,Reaction) :-
    template(Action1,Reaction1),
    match(Action1,Action),
    match(Reaction1,Reaction),
    !.

template([s([i,am]),s(X)],
         [s([why,are,you]),s(X),w('?')]).
template([s([i,heard]),s(X)],
         [s([where,did,you,hear]),s(X),w('?')]).
template([s([i,feel]),s(_)],
         [s([do,you,often,feel,that,way]),w('?')]).
template([w(i),s(X),w(you)],
         [s([why,do,you]),s(X),w(me),w('?')]).
template([s([where,is]),s(X)],
         [s(X),s([is,on,aisle,13]),w('.')]).
template([s([i,saw]),s(X)],
         [s([when,did,you,see]),s(X),w('?')]).
template([s([i,eat]),s(X)],
         [s([do,you,often,eat]),s(X),w('?')]).
template([s([bye])],
         [s([goodbye,and,have,a,nice,day]),w('.')]).
template([s([why])],
         [s([because,i,am,concerned,for,you]),w('!')]).
template([s(_)],
         [s([please,go,on])]).

match([],[]).
match([Item|Items],[Word|Words]) :-
    match(Item,Items,Word,Words).

% The following match/4 predicates are really
% just basic constraints we'd ordinarily expect.

% Word = Word is trivially true, of course,
% and we're requiring that Items and Words can
% be unified.
match(w(Word),Items,Word,Words) :-
    match(Items,Words).

% Here, the variable names tell the story.
match(s([Word|Rest_Of_Segment]),Items,Word,Words) :-
    append(Rest_Of_Segment,Words_Net_Of_Rest_Of_Segment,Words),
    match(Items,Words_Net_Of_Rest_Of_Segment).
```