```
%Michael Alsbergas, 5104112
cities([liverpool,preston,manchester,lancaster,carlisle,leeds]).
testers([vincent,mia,butch,zed,marcellus,jules]).

%link/2 records all the trains that travel from 1 city to another.
link(liverpool,preston).
link(liverpool,manchester).
link(preston,lancaster).
link(preston,manchester).
link(lancaster,carlisle).
link(lancaster,leeds).
link(carlisle,leeds).
link(manchester,leeds).
%tester links
link(vincent,mia).
link(vincent,butch).
link(butch,zed).
link(zed,marcellus).
link(marcellus,mia).
link(jules,marcellus).
link(jules,vincent).

%direct/2 returns a list of all cities directly linked to city C.
connect(_,L,[],L).

connect(C,L,[H|T],X):- link(C,H),
        connect(C,[H|L],T,X).

connect(C,L,[H|T],X):- link(H,C),
        connect(C,[H|L],T,X).

connect(C,L,[_|T],X):- connect(C,L,T,X).

direct(C,X):-connect(C,[],
                    [liverpool,preston,mancester,lancaster,carlisle,leeds,vi ce
nt,mia,butch,zed,marcellus,jules]X).

%inlist checks to see if X is an element of the list.
inlist(X,[X|_]).
inlist(X,[_|Y]):- inlist(X,Y).

%merge combines 2 lists together and removes doubles.
combine([],[],L,L).

combine([H|T],[],L,X):-
        inlist(H,L),
        combine(T,[],L,X).

combine([H|T],[],L,X):- combine(T,[],[H|L],X).

combine(A,[H|T],L,X):-
        inlist(H,L),
        combine(A,T,L,X).

combine(A,[H|T],L,X):- combine(A,T,[H|L],X).

merge(A,B,X):- combine(A,B,[],X).

% addcities uses direct and merge to expand the list of possible
% destinations.
addcities([],Y,Y).
addcities([H|T],A,Y):-
        direct(H,Z),
```

```
        merge(A,Z,B),
        addcities(T,B,Y).


%linked returns the list of possible destinations within N transfers.
transfer(X,0,X).

transfer(C,N,X):-
        addcities(C,C,Y),
        M is N-1,
        transfer(Y,M,X).

linked(C,N,X):- transfer([C],N,X).
```