[Week 2 - Lab Solutions!](https://www.ole.bris.ac.uk/webapps/blackboard/content/listContent.jsp?course_id=_257207_1&content_id=_8044794_1&mode=reset)

Preparatory Tasks:

* % get agent starting position
  + ?- my\_agent(A), get\_agent\_position(A,P).
* % define posible move directions (spelt out here in full for clarity)
  + m(north). m(east). m(south). m(west).
* % define valid on grid positions
  + on\_board(p(X,Y)) :- ailp\_grid\_size(N), between(1,N,X), between(1,N,Y).
* % define possible steps (without bounds checking)
  + pos\_step(p(X,Y),north,p(X,Y1)) :- Y1 is Y-1.
  + pos\_step(p(X,Y),east,p(X1,Y)) :- X1 is X+1.
  + pos\_step(p(X,Y),south,p(X,Y1)) :- Y1 is Y+1.
  + pos\_step(p(X,Y),west,p(X1,Y)) :- X1 is X-1.
* % define new positions (with bounds checking)
  + new\_pos(P,D,Q) :- on\_board(P), pos\_step(P,D,Q), on\_board(Q).
* % see if spiral is complete by checking if all cells are in path
  + complete(L) :- forall(on\_board(P),member(P,L)).
* % Note this is more thorough than the simpler alternative
  + % complete(L) :- ailp\_grid\_size(N), N2 is N\*N, length(L,N2).

Spiral Generation

* % define a clockwise turn
  + turn(north,east,clockwise).
  + turn(east,south,clockwise).
  + turn(south,west,clockwise).
  + turn(west,north,clockwise).
* % define an anticlockwise turn
  + turn(D1,D2,anticlockwise) :- turn(D2,D1,clockwise).
* % Base case: spiral complete
  + spiral(Ps,Qs,\_) :- complete(Ps), !, reverse(Ps,Qs).
* % Recursive case: continue straight or turn
  + spiral([Q,P|Ps],Qs,S) :- new\_pos(P,D,Q), (C=D ; turn(D,C,S)), new\_pos(Q,C,R), \+ member(R,[P|Ps]), spiral([R,Q,P|Ps],Qs,S).
* % Wrapper predicate
  + spiral(Ps) :- my\_agent(A), get\_agent\_position(A,P), new\_pos(P,\_,Q), spiral([Q,P],Ps,\_).
* % Example query
  + ?- spiral([\_|Ps]), my\_agent(A), agent\_do\_moves(A,Ps).