2010 Paper 3 Question 7

Prolog

(*a*) Consider the following clauses:

```
a(1).
a(a).
b(2).
b(3).
c(X,X) :- a(X).
c(X,Y) :- a(X),!,b(Y).
c(X,X) :- b(X).
```

List all of the possible solutions to the query `c(A,B)` in order. [2 marks]

(*b*) Consider the predicate `p/4` defined as:

```
p(_,[],[],[]).
p(P,[X|Xs],[X|L1],L2) :- X < P, p(P,Xs,L1,L2).
p(P,[X|Xs],L1,[X|L2]) :- X >= P, p(P,Xs,L1,L2).
```

(*i*) What does `p/4` do? Show an example query and response. [2 marks]

(*ii*) What types of terms cannot be used as the first `p/4` argument? [2 marks]

(*iii*) Describe where and why red and green cuts can be usefully employed within the `p/4` predicate, modifying the predicate if necessary. [3 marks]

(*iv*) The quicksort algorithm requires selection of a pivot value, forming two sublists, and then building the sorted list from the results of recursively quicksorting the sublists. The first sublist contains all elements that are less than the pivot value. The second contains the remaining elements.

Implement quicksort in a predicate `qs(+In,-Out)` that binds `Out` to a sorted version of the list `In`, and uses the `p/4` predicate above. Assume that `In` only contains numbers. Do not optimise your choice of pivot value: just use the head of the input list. [5 marks]

(*c*) The predicate `flatten(+In,-Out)` is defined to expect that `In` be unified with a list that may contain elements that are themselves lists (and likewise for those lists, up to any depth). In such cases, `Out` is unified with a list that never has elements that are lists—all lists are expanded in place. For example, `flatten([1,2,[3,4],[[5],6]],[1,2,3,4,5,6])` would be true.

Implement the `flatten/2` predicate. When given a valid query, your solution should not provide any spurious results. [6 marks]