1. Here is Prolog code for quicksort:

```prolog
qsort([],[]).
qsort([X|Y],Z) :- partition(X,Y,Y1,Y2),
                qsort(Y1,Z1), qsort(Y2,Z2),
                append(Z1,[X|Z2],Z).

append([],W,W).
append([A|B],W,[A|V]) :- append(B,W,V).

partition(_,[],[],[]). (*)
partition(X,[A|B],[A|Y1],Y2) :- A < X, partition(X,B,Y1,Y2).
partition(X,[A|B],Y1,[A|Y2]) :- A >= X, partition(X,B,Y1,Y2).
```

(a) In the line marked (*), what is ‘_’? Can we use a free variable instead of ‘_’?

(b) Is the above program reversible, i.e., on giving an unsorted list as the first argument it binds
the second argument to a sorted list, but on giving a sorted list as the second argument
does it bind the first argument to all possible permutations of the elements? If not, explain
why not.

(c) Implement quicksort in a different manner which does show reversible input-output behav-
ior.

2. Here is Prolog code to generate all permutations of a set of items:

```prolog
remove(X,[X|Xs],Xs).
remove(X,[Y|Ys],[Y|Zs]) :- remove(X,Ys,Zs).

permute(Xs,Ys) :- mystery(Xs,Ys,Ys).

mystery([],[],[]).
mystery([X|Xs],Ys,[|Zs]) :- mystery(Xs,Ys,Zs), remove(X,Ys1,Ys).
```

(a) Consider only the two rules for remove. What happens if there are multiple occurences
of an item in a list? Are all the occurences of the item removed at once or are the lists
outputted with X removed one at a time and the others intact?

(b) Explain clearly what mystery is meant to do.