8 Prolog (DE)

(*a*) Give the result and any variable bindings that occur from making each of the following (independent) queries.

(*i*) `32 = A.` [1 mark]

(*ii*) `a(b(6, A)) = a(b(B, 2)).` [1 mark]

(*iii*) `a(b(x, A)) = a(b(5, 4)).` [1 mark]

(*iv*) `A = 3 + 6, A is 9.` [1 mark]

(*v*) `A is 3 + 6, A = 9.` [1 mark]

(*b*) Given the following clauses,

```
a(4).
a(x).
b(3,x).
b(1,7).
c(A, B, C) :- a(A), b(B, _), !, a(D).
c(A, _, B) :- b(A, B).
```

(*i*) List the solutions reported by Prolog to the query `c(P, Q, R)`, for each giving any binding of variables that occurs. [2 marks]

(*ii*) Explain whether the query in part (*b*)(*i*) can bind `P` to `x`. [1 mark]

(*iii*) List the solutions to the query `c(1, 6, R)`, for each giving any binding of variables that occurs. [1 mark]

(*c*) We represent a binary tree using a term of the form `[Left,NodeName,Right]` or `null`. Here is an example perhaps recording a tree of excursions between countries:

```
[[null,de,null],uk,[[null,ua,null],pt,null]]
```

Write a predicate `p(+Tree, +NNames, -Ps)` where `Tree` is a tree as above, `NNames` is a list of node names, and `Ps` is to be unified with a list of elements of the form `[N,P]` where `N` is in `NNames` and `P` is `N`'s parent in `Tree`. [For example, supposing `T` is bound to the above tree, the query `p(T, [ua, za], Ps)` binds `Ps` to `[[ua, pt]]`.] [5 marks]

(*d*) Difference lists are a powerful tool for increasing efficiency but they address a very specific problem. Does this issue arise in your implementation of `p/3`? In other words, can `p/3` be made more efficient using difference lists and why (or why not)? [2 marks]

(*e*) Regardless of your answer above, write a predicate `pdl/3` that behaves just like `p/3` but uses difference lists in its implementation. [4 marks]