1. Answer the following three parts.

   a. Give a Prolog procedure which implements a predicate prefix(X, Y), meaning that X is an initial sublist of a list Y.

      (You must fully define all the predicates you use unless they are built-in.)

      An example of querying prefix is:

      prefix(X, [a,b,c]).

      X = []; X = [a]; X = [a,b]; X = [a,b,c]; no

      [11 marks]

   b. Give a Prolog procedure which implements a predicate suffix(X, Y), meaning that X is a terminal sublist of a list Y.

      (You must fully define all the predicates you use unless they are built-in.)

      An example of querying suffix is:

      suffix(X, [a,b,c]).

      X = [a,b,c]; X = [b,c]; X = [c]; X = []; no

      [11 marks]

   c. Explain the meaning of the following procedure for the predicate mystery.

      mystery([], X).

      mystery(X, [Y | Z]) :- mystery(X, Z).

      mystery([X | Y], [X | Z]) :- mystery(Y, Z).

      Give two example queries for the mystery procedure and their answers; one of the queries should succeed and the other should fail.

      [11 marks]

[Total 33 marks]

2. Answer the following three parts.

   a. Write a Prolog program that states that birds normally fly unless they are exceptional. As an example eagles are birds that fly, but penguins are exceptional birds.

   [11 marks]

   b. Explain the meaning of the following procedure for the predicate puzzle.

   ```
   puzzle([], L, L) :- !.
   puzzle([H | T1], L2, [H | T3]) :- puzzle(T1, L2, T3).
   ```

   Is the cut in the predicate puzzle green or red ? Justify your answer with an example query.

   [11 marks]

   c. Write a Prolog procedure implementing a predicate if-then-else(P, Q, R), meaning that if P is true then do Q else do R.

   [11 marks]

   [Total 33 marks]

3. Answer the following three parts.

   a. Write a Prolog procedure implementing a predicate go(X, Y, R), meaning that if we can go through a door from room X to room Y, given that R is a list of rooms that we have already visited.

      Assume that a predicate door(X, Y) is available that specifies that there exists a door from room X to room Y and from Y to X.

      [11 marks]

   b. Modify part (a) of the question such that on our way from room X to room Y we must visit a room that has a terminal and a phone, or a room that has a terminal and a fax.

      [11 marks]

   c. Consider the predicate member(X, Y) meaning that X is a member of the list Y, and also consider the **not** predicate implementing negation by failure.

      What is the difference between the following two queries ?

      ? member(X, [a, b, c]), write(X).

      ? not(not(member(X, [a, b,c]))), write(X).

      [11 marks]

[Total 33 marks]

4. Answer the following five parts.

a. The order of expressions in logic is unimportant if different versions of these expressions have the same logical meaning. For example, if A+B or A.B. are ways of writing “A and B”, then B+A or B.A. are equally acceptable.

What is the situation for the corresponding issue in Prolog, and why ?

[8 marks]

b. Write Prolog code to express the following information, with the order of items in the English text below being strictly reproduced in your answer:

“Somebody is a person if there is some other person who is the mother of that person. Eve is a person”.

[6 marks]

c. What happens when the program fragment from (b) is used to determine whether or not you are a person ?

How can any of the shortcomings of the program’s behaviour be removed ?

[5 marks]

d. If alist(L) is a predicate that succeeds if and only if L is a list, a possible definition is

```
alist([X | Y]) :- alist(Y).
alist([]).
```

What will this predicate do if we try to evaluate

(i) ? alist([a,b]).

(ii) ? alist(L).

[6 marks]

e. Someone claims that the definition of the predicate in (d) would be better as

```
alist([]).
alist([_|_])
```

and somebody else says that this would be even better if the order of those two lines were reversed. State your opinion of the two claims, and justify it.

[8 marks]

[Total 33 marks]

5. Answer the following four parts.

   a. A popular interpretation of what "Prolog" means is "programming in logic".

      How correct is this interpretation ? Justify your answer.

      [6 marks]

   b. Name any two other interpretations of what kind of computing occurs when a Prolog program runs. For each, suggest and describe an example of a computation that is best viewed in that interpretation.

      [8 marks]

   c. Suppose that a file is available with a sequence of assertions of the form course(label, day, time, place), where each one gives information on where and when one weekly lecture, tutorial. etc. of a particular UCL course takes place.

      Write a Prolog program which takes as data a list of the labels of the courses in which you may think of enrolling and which is able to compute a list of any conflicts in your desired timetable. Explain in English the key design choices in your answer.

      [14 marks]

   d. Comment on how well or badly Prolog is suited for the writing of a good program to solve the problem in (c).

      [5 marks]

   [Total 33 marks]

6. Answer the following four parts.

   a. Describe briefly the minimax method of computing a best next move in a game which two players make their moves in alternation.

   [5 marks]

   b. Design and write an outline Prolog program to determine a best next move by the minimax method, amd explain your design. "Outline" means a top-down design, breaking the overall problem down into sub-problems, giving appropriate Prolog functors that solve the sub-problems, and writing these in the correct order in the clauses that use them. Do not go into any detail on (i) the choice of data structures and (ii) the way in which the program chooses the next game position to inspect.

   [15 marks]

   c. Describe how you might turn the design in (b) into an actual program by taking the steps (i) and (ii) in (b), for a board game such as noughts and crosses or tic-tac-toe. Mention any difficulties of programming these steps.

   [8 marks]

   d. Suppose that the game is not a simple board game but a game (like diplomacy or politics) where there is no fixed set of possible moves. Instead, assume that for stage S of the game there is a generator of possible moves which will give a different move on each occasion when it is called, or fail if it has already generated all the moves that are available in S.

   Indicate any changes that should then be made to your answer for (b) and (c), and explain why the changes should be made.

   [5 marks]

[Total 33 marks]

END OF PAPER