Question 1.

Write the following program and compile it:

```
% Program:  ROYAL

parent(queenmother,elisabeth).   parent(elisabeth,charles).
parent(elisabeth,andrew).        parent(elisabeth,anne).
parent(elisabeth,edward).        parent(diana,william).
parent(diana,harry).             parent(sarah,beatrice).
parent(anne,peter).              parent(anne,zara).
parent(george,elisabeth).        parent(philip,charles).
parent(philip,andrew).           parent(philip,edward).
parent(charles,william).         parent(charles,harry).
parent(andrew,beatrice).         parent(andrew,eugene).
parent(mark,peter).              parent(mark,zara).
parent(william,georgejun).       parent(kate,georgejun).
parent(william,charlotte).       parent(kate,charlotte).
parent(phillip,anne).            parent(william,louis).
parent(kate,louis).
```


Define the following predicates on the persons in the program `ROYAL`.

(1a) `the_royal_females/1` (the argument is a list of all female members of the Royal Family)

(1b) `the_royal_males/1` (the argument is a list of all male members of the Royal Family)

(2) `the_royal_family/1` (use (1a) and (1b))

(3) `mother/2`

(4) `ancestor/2`

(5) `sibling/2`

(6) `brother/2`

Translate the following questions into Prolog queries and try them out:

(7) Who is an ancestor of Louis?

(8) Who is a grandchild of Queenmother? (Define a predicate `grandmother` to answer this question.)

(9) Who are the common descendants of Anne and Edward? (Define a predicate `common_anchestors`.)

(10) Who has a brother who is grand dad? (Define a predicate

`has_brother_who_is_granddad`.)

.

[10 marks]

Question 2. Define the following predicates on lists:

a) Write a predicate `nth_elt(N,L,E,R)` which selects the `N`th element, `E` from a list `L` leaving list `R`. Also define `nth_elt_with_test(N,L,E,R)` which also checks that `N` is >0 and ≤ length of `L`.

b) Write a predicate `sort(L,LS` that sorts a list (L). [Note, an experienced logic programmer would use a library function for that, it is nevertheless a good exercise to program that on your own.]

c) Finally use a) and b) to define `median(L,M)` which finds the median `M` of a list `L`.

[10 marks]

Question 3. An example of a recursive predicate and a tail recursive version using an accumulating parameter.

a) The square of the Euclidean distance between two vectors x_i and y_i is $\Sigma_{i=1}^{n}(x_i - y_i) * (x_i - y_i)$. Write a recursive predicate `euclidsqr(X,Y,ED)` which returns the value in `ED` when `X` and `Y` are lists representing vectors of the same length.

b) Now write a tail recursive predicate `euclidsqr_acc(X,Y,A,ED)` to compute the same function using the accumulating parameter `A` to store intermediate calculations. (Look at `sum_a` to be discussed in lectures).

[10 marks]

Question 4. (Backtracking Solution of Sujiko Puzzle)

As the caption says, the aim is to place distinct digits from 1-9 in the empty cells so that all the digits are used (including the given digits) and the numbers in the circles are the sum of the *four* surrounding digits.

You are to write a generate and test backtracking program in Prolog to solve these two puzzles as follows:

a) Define the predicate `member_rem(E,L,R)` which chooses an element `E` from list `L` leaving remainder `R`.

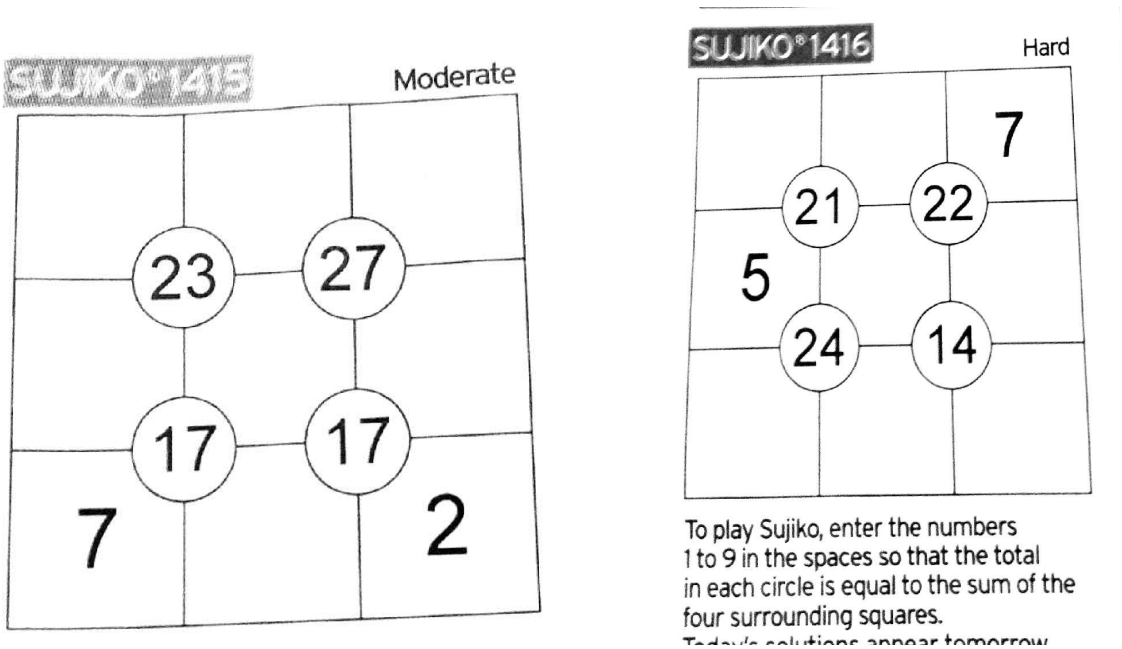

Figure 1: Sujiko Puzzles, moderate and hard

b) Using the above define `gen_list(N,L,D)` which generates a list `L` of `N` *distinct* elements from the list `D` where the length of `D` is $\geq$ `N`.

c) Write two predicates `solve_sujiko1([X1,X2,X3,X4,X5,X7])` and `solve_sujiko2([X1,X2,X3,X4,X5,X7])` which solve the two puzzles given above, where `X1` – `X7` will be the values of the digits placed in the empty cells (left to right, top to bottom order). This is done by generating a possible solution and then testing if all the constraints of the puzzles are satisfied.

Include an example run of your puzzles as comment.

d) Write an improved generalized version

```
new_solve_sujiko2([X11,X12,X22,X23,X31,X32,X33])
```

that avoids generating all possible solutions, but solves the puzzle step-by-step.

[10 marks]

Quality of Work, Programming Style, Presentation of code and comments, Adherance to instructions, Elegance.

[10 marks]