

CSC 600-01 (SECTION 1)
Homework 3 - Logic Programming in Prolog
prepared by Ilya Kopyl

CSC 600 HOMEWORK 3 - LOGIC PROGRAMMING IN PROLOG

March 13, 2018

*Homework is prepared by: Ilya Kopyl.
It is formatted in LaTeX, using TeXShop editor (under GNU GPL license).
Diagrams are created in LucidChart online editor (lucidchart.com).*

1. Write a PROLOG program that investigates family relationships using lists. The facts should be organized as follows:

```
m([first_male_name, second_male_name, ..., last_male_name]).  
f([first_female_name, second_female_name, ..., last_female_name]).  
family( [father, mother, [child1, child2, ..., child_n]] ).
```

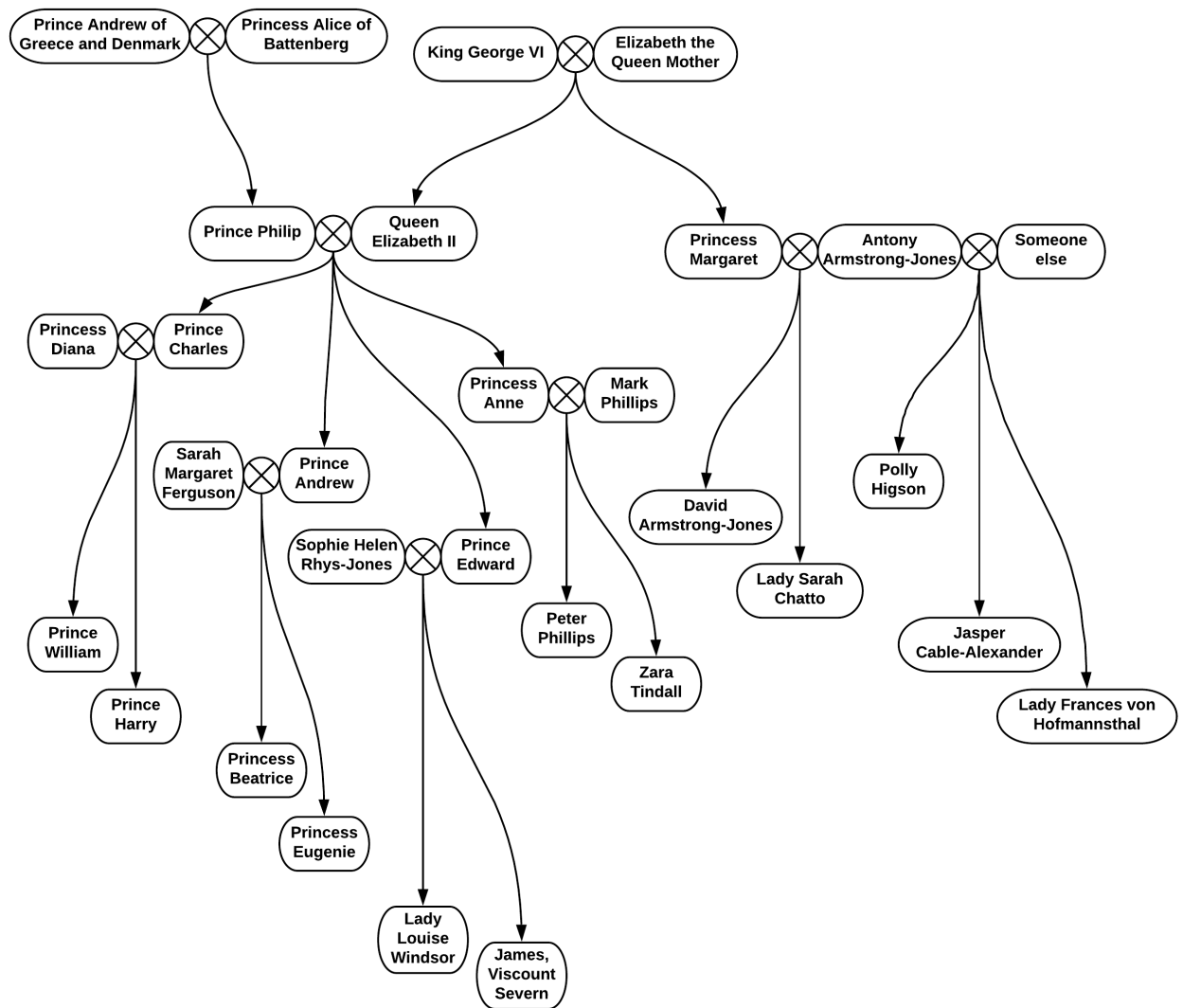
Write rules that define the following relationships:

```
male(X)  
female(X)  
father, mother, parent  
siblings1, siblings2  
brother1, brother2  
sister1, sister2  
cousins  
uncle, aunt  
grandchild, grandson, granddaughter  
greatgrandparent  
ancestor
```

For each of these rules show an example of its use.

The answer is listed on the pages TBD through TBD.

Partial diagram of the family tree of the British Royal family.



The code listing of maxlen function:

```
somePredicate(A, B) :-  
    arbitraryPredicate(A, _, 1, 2),  
    predicateWithAtom(someAtom),  
    anotherPredicate(B, someAtom, myPredicate(A, _)),  
    findall(X, ('testString'(X), myPredicate(A, X)), L1),  
    member(A, L1),  
    !.
```

The code listing of main program:

```
somePredicate(A, B) :-  
    arbitraryPredicate(A, _, 1, 2),  
    predicateWithAtom(someAtom),  
    anotherPredicate(B, someAtom, myPredicate(A, _)),  
    findall(X, ('testString'(X), myPredicate(A, X)), L1),  
    member(A, L1),  
    !.
```

The result of the program execution:

Standard output:

2. Write a PROLOG program that includes the following operations with lists:

membership testing (is an element member of a list?)
first element
last element
two adjacent elements
three adjacent elements
append list1 to list2 producing list3
delete element from a list

append element to a list
insert element in a list
compute the length of list
reverse a list
check whether a list is a palindrome
display a list

For each of these operations write your implementation of the operation and show an example of its use. If a predicate already exists (predefined in Prolog), modify its name (e.g. myappend or append1). Lists to be processed can be created by an auxiliary program, defined as facts, or entered from the keyboard.

The answer is listed on the pages TBD through TBD.

The code listing of the two-dimensional array that stores bit pattern of each BigInt digit. It is declared in the global space (outside of any function).

```
somePredicate(A, B) :-  
    arbitraryPredicate(A, _, 1, 2),  
    predicateWithAtom(someAtom),  
    anotherPredicate(B, someAtom, myPredicate(A, _)),  
    findall(X, ('testString'(X), myPredicate(A, X)), L1),
```

```
member(A, L1),  
!.
```

Main program, excluding the declaration of BIG_DIGITS array:

```
somePredicate(A, B) :-  
    arbitraryPredicate(A, _, 1, 2),  
    predicateWithAtom(someAtom),  
    anotherPredicate(B, someAtom, myPredicate(A, _)),  
    findall(X, ('testString'(X), myPredicate(A, X)), L1),  
    member(A, L1),  
    !.
```

3. Write a PROLOG program that solves the 8 queens problem (location of 8 queens on a chess board so that no queens have each other in check, i.e. are not located in the same row/column/diagonal).

The answer is listed on the pages TBD through TBD.