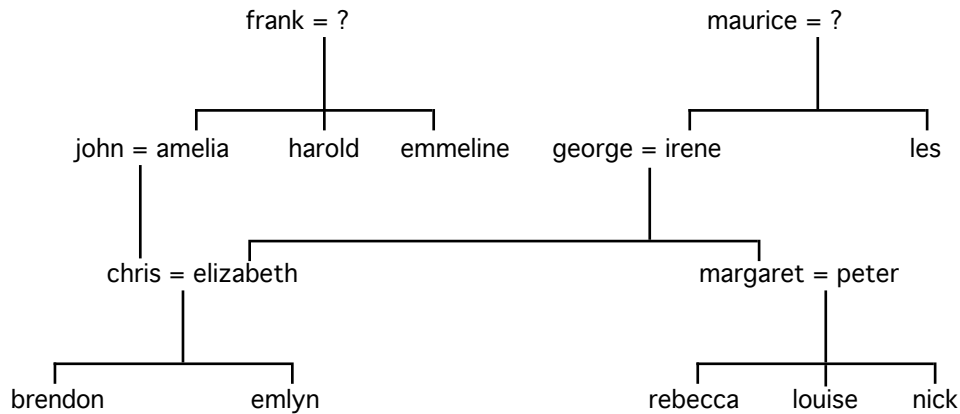


CM20019-Prolog Lab part 1

The aim of this exercise is to familiarize you with simple Prolog programs and queries, and is based entirely upon a database representing this fragment of a family tree:



The database is just a set of atomic facts and will be already implemented so that you can use it straightaway. It uses just three predicates:

<code>child_of(X, Y)</code>	"X is a child of Y"
<code>male(X)</code>	"X is male"
<code>female(X)</code>	"X is female"

For instance, the first fact in the database is

```
child_of(emmeline, frank).
```

and expresses "emmeline is a child of frank".

1. Enter the following queries and find all their solutions:

```
?- child_of(peter, irene).
?- child_of(peter, emlyn).
?- child_of(X, george).
?- child_of(george, Y).
?- child_of(X, X).
?- child_of(X, Y).
```

2. Add further Prolog clauses which define the following predicates:

<code>daughter_of(X, Y)</code>	"X is a daughter of Y"
<code>uncle_of(X, Y)</code>	"X is an uncle of Y"
<code>niece_of(X, Y)</code>	"X is a niece of Y"
<code>grandfather_of(X, Y)</code>	"X is a grandfather of Y"

Test each one by posing suitable queries and finding all their solutions. As a guiding example, this is how we might define "X is a sister of Y":

```
sister_of(X, Y) :- child_of(X, Z), child_of(Y, Z), X \== Y, female(X).
```

Note that in Prolog `\==` (backslash, equals, equals) means "not identical to".