1. 使用语言：prolog（swi-prolog）

2. 过河问题：

假设初始时人和船都在左岸，过河后都在右岸。

将过河过程中人员及小船在两岸的分布情况对应到一个状态，这里我用一个 list 来表示，list 中的元素代表当前在左岸的人或船，例如，初始状态为[policeman, criminal, father, mother, son1, son2, daughter1, daughter2, boat]， 末状态为[]。

将不安全的状态根据要求用谓词 unsafe（State）表示出来。将每次过河时，用来表示状态的 list 会做出更新，这里更新时要考虑到只有一部分人是可以划船的，该过程用谓词 move（State，NewState）表示。

最后，通过类似于递归的过程，将上一步 move 产生的 NewState 作为下一步的初始的 State，每移动一次后判断当前状态是否安全，且该状态是否已经出现过。直到 move 产生的新状态为末状态时结束。

运行方式：双击 crossingRiver.pl 文件，输入 start.

产生输出如下（每个 list 代表一个状态）：

```
1 ?- start.
Path:
[policeman,criminal,father,mother,son1,son2,daughter1,daughter2,boat]
[father,mother,son1,son2,daughter1,daughter2]
[policeman,boat,father,mother,son1,son2,daughter1,daughter2]
[father,mother,son2,daughter1,daughter2]
[criminal,policeman,boat,father,mother,son2,daughter1,daughter2]
[criminal,policeman,mother,daughter1,daughter2]
[father,boat,criminal,policeman,mother,daughter1,daughter2]
[criminal,policeman,daughter1,daughter2]
[mother,boat,criminal,policeman,daughter1,daughter2]
[mother,daughter1,daughter2]
[father,boat,mother,daughter1,daughter2]
[daughter1,daughter2]
[mother,boat,daughter1,daughter2]
[daughter2]
[criminal,policeman,boat,daughter2]
[criminal]
[policeman,boat,criminal]
[]

true
```

答案解释：

全在左岸，
Policeman 带 criminal 到右岸，
Policeman 独自回到左岸，
Policeman 带 son1 到右岸，
Policeman 带 criminal 回到左岸，
Father 带 son2 到右岸，
Father 独自回到左岸，
Father 带 mother 到右岸，
Mother 独自回到左岸，
Policeman 带 criminal 到右岸，

Father 独自回到左岸，
Father 带 mother 到右岸，
Mother 独自回到左岸，
Mother 带 daughter1 到右岸，
Policeman 带 criminal 回到左岸，
Policeman 带 daughter2 到右岸，
Policeman 独自回到左岸，
Policeman 带 criminal 到右岸。

3. Einstein's puzzle

用 Mans = [man(N1, C1, D1, P1, S1), man(N2, C2, D2, P2, S2), man(N3, C3, D3, P3, S3), man(N4, C4, D4, P4, S4), man(N5, C5, D5, P5, S5)] 来表示这五个人按照顺序的组合方式，其中 man(N，C，D，P，S)表示某一个人。

将所有给出的条件都放在一个谓词 rule(Mans)里，用合取的方式连接，这样就可以就可以得到满足条件的 Mans。再在 Mans 中找出包含 fish 的 man 即可。

运行方式：双击 Einstein.pl，输入 start.
输出结果：

```
1 ?- start.
German
true
```

答案解释：养鱼的是德国人。

4. 车主问题：

与前一个问题类似，区别在于对于每个人所说的话有两种情况，说谎和未说谎。两种不同的情况用析取的方式连接。

运行方式：双击 cars.pl，输入 start.
输出结果：

```
1 ?- start.
[driver(George,Chevrolet,25),driver(Doc,Dodge,15),driver(Tito,Toyota,20),driver(Jimmy,Ford,30)]
true
```

答案解释：
George：Chevrolet，25
Doc：Dodge，15
Tito：Toyota，20
Jimmy：Ford，20