

**SC3000 Artificial Intelligence**

**Lab Group SDAB**

**Team Name: AI Triad**

Team Members:

|  |  |  |
| --- | --- | --- |
| **Full Name** | **Matriculation Number** | **Individual Contribution** |
| Keh Jing Xiang | U2320460L | Discussions, Documentation,  Prolog Programming |
| Chow Weng Shi | U2320760B | Discussions, Documentation,  Prolog Programming |
| Choo Zhen Ming | U2320950J | Discussions, Documentation,  Prolog Programming |

# Exercise 1: The Smart Phone Rivalry

## **1. Translate the natural language statements above describing the dealing within the Smart Phone industry into First Order Logic (FOL).**

Competitor(sumsum, appy) - sumsum is a competitor of appy.

Competitor(appy, sumsum) - appy is a competitor of sumsum.

SmartphoneTech(galactica\_s3) - galactica-s3 is a Smart Phone Technology.

Developed(sumsum, galactica\_s3) - sumsum developed galactica-s3

Stole(stevey, galactica\_s3) - stevey stole galactica-s3.

Boss(stevey, appy) - stevey is a boss of appy.

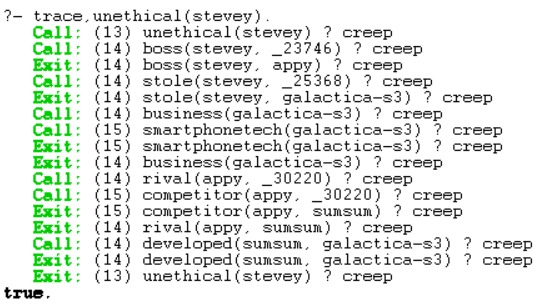
∀A,∀B, (Competitor(A, B) → Rival(A, B)) - All competitors are rivals.

∀X, (SmartphoneTech(X) → Business(X)) - All Smart Phone Technology are Business.

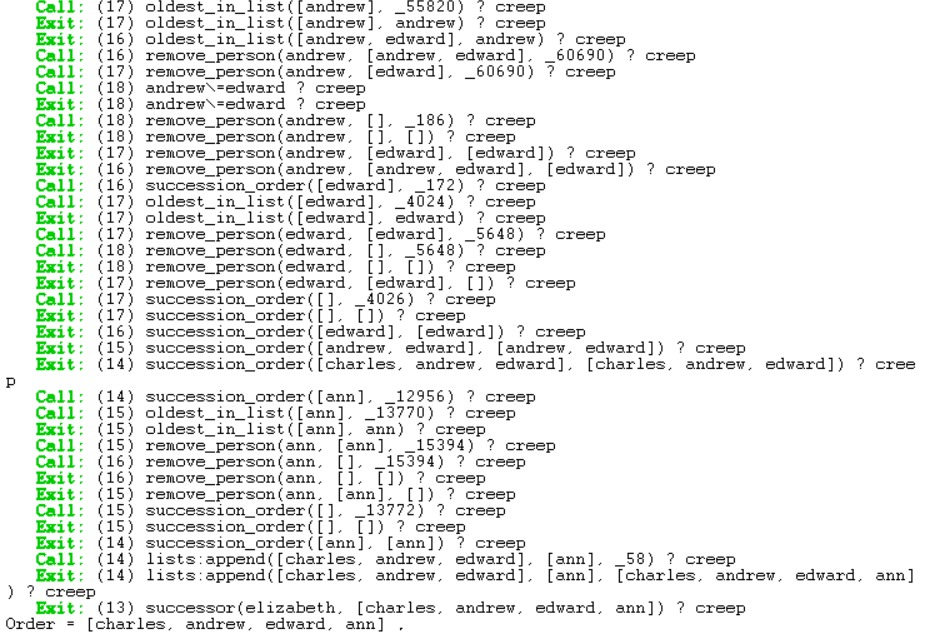
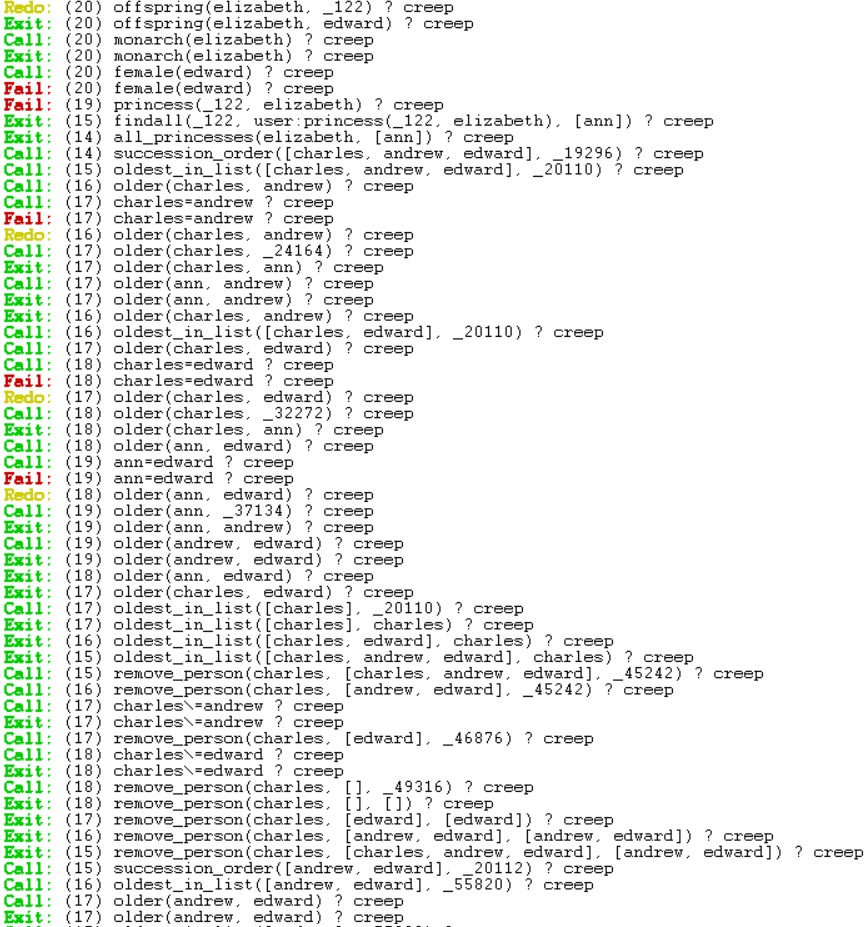
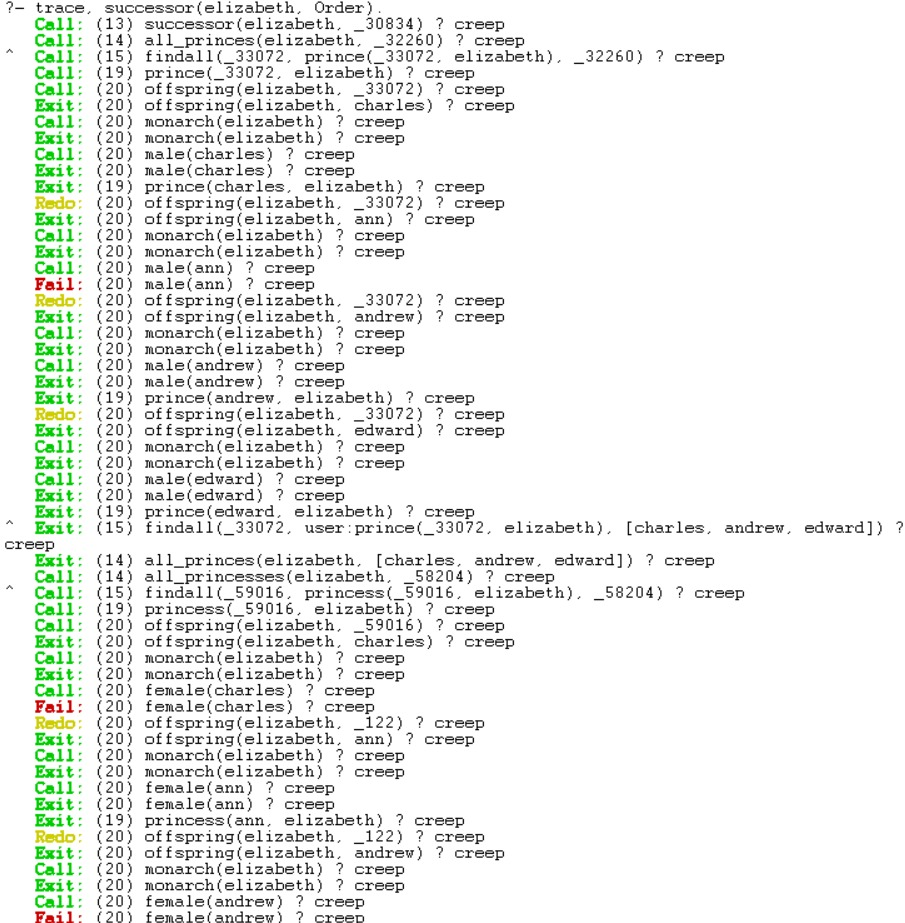
∀A, ∀B, ∀C,∀D, [ Boss( A, B)) ∧ Stole(A, D) ∧ Business(D) ∧ Developed(C, D) ∧ Rival(B, C) → Unethical(A) - If a boss of a company steals a business developed by a rival company, then the boss is unethical.

## **2. Write these FOL statements as Prolog clauses.**

Please refer to the file uploaded.

**3. Using Prolog, prove that Stevey is unethical. Show a trace of your proof. ethical. Show a trace of your proof.**

# Exercise 2: The Royal Family

1. **Code Explanation:**  
   The princes are added into a list, **A**, containing only princes. The princesses are added into a separate list, **B**, containing only the princesses. We then search through list **A** to find the oldest prince, remove it from **A** then add it into a new list, **A2** from the front. Continue doing this until A is empty. Do the same for princesses until **B** is empty and **B2** is full of the correctly ordered princesses from oldest to youngest. Then finally we append **A2** and **B2** together.  
     
   **Trace:**  
   
2. **Code Explanation for changes made:**  
   Similar to Exercise 2.1, but instead of having separate lists A and B containing the princes and princesses, we now have a function to just find all the descendants and put them into one single list **X**. Then we repeatedly find the oldest from **X**, remove it then append it to the front of a new list, **Y**, until list **X** is empty. Thus, **Y** now contains the order of the descendants from oldest to youngest regardless of the age.  
     
   **Trace:**  
   