Answer 3 of the following 5 questions for 7 points each.

Q1. Write a Prolog or Haskell program that recognizes if a string made of left and right parentheses is "well balanced" (as shown in the following examples):

<table>
<thead>
<tr>
<th>String</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;()&quot;</td>
<td>true</td>
</tr>
<tr>
<td>&quot;())&quot;</td>
<td>false</td>
</tr>
<tr>
<td>&quot;((()&quot;</td>
<td>false</td>
</tr>
<tr>
<td>&quot;()()&quot;</td>
<td>true</td>
</tr>
<tr>
<td>&quot;(()())()&quot;</td>
<td>true</td>
</tr>
<tr>
<td>&quot;()()(&quot;</td>
<td>false</td>
</tr>
</tbody>
</table>

Q2. Using the following Prolog program as an executable specification

```prolog
% ksubset(K,Xs, Rs): computes subsets Rs of K elements of the set Xs
ksubset(0,_,[]). ksubset(K,[X|Xs],Rs):-K>0,K1 is K-1,ksubset(K1,Xs,Rs). ksubset(K,[|Xs],Rs):-K>0,ksubset(K,Xs,Rs).
```

write an equivalent Haskell program.

Q3. Write a Prolog program that solves the following arithmetic puzzle

```
   A D A M
  + A N D
  + E V E
  ---------
  M O V E D
```

where each letter represents a unique digit from 0 to 9 and the value below the line represents the sum of those above it.

Q4. A Fibonacci-n sequence is defined by having the first n values equal to 1 and then the each next value the sum of the previous n ones. For n=3, implement a Haskell function that can compute in less then a minute the value of the 100-th Fibonacci-3 number.

Q5. Using the family.pro Prolog program defined in class add the following predicates: - second
cousin (defined as a first cousin of a first cousin that’s not a first cousin, a sibling or self) - great aunt (defined as the aunt of a parent) - great uncle (defined as the uncle of a parent)

**Answer the following question: (4 pts).**

**Q6.** Name the most unpleasant feature of Prolog or Haskell that you would suggest to be removed from the language. Explain briefly why.