Answer 2 of the following 3 questions, for 15 points each:

Q1. Write short Python program that you think would be difficult to express in Haskell or Prolog. Explain what makes writing the equivalent Haskell or Prolog code complicated, awkward or tedious.

Q2. Write a Haskell or Prolog program equivalent to the following Python program:

```python
# a tree is either an integer leaf or a tuple of children trees
def collectLeaves(tree):
    
def yieldLeaf(tree):
        if isinstance(tree,int): yield tree
        else:
            for child in tree:
                for leaf in yieldLeaf(child):
                    yield leaf
    
    return list(yieldLeaf(tree))
```

Here are some examples of use – on which your Haskell or Prolog program should return the same result.
```python
>>> collectLeaves( ((1,2),((3,4,5),(6,7))) )
[1, 2, 3, 4, 5, 6, 7]
>>> collectLeaves( (0,((3,1),0),((3,9,42), (8,0)),100) )
[0, 3, 1, 0, 3, 9, 42, 8, 0, 100]
```
Q3. The following Prolog program finds all colorings of a graph with a list of available colors:

Write an equivalent Haskell or Python program and show its execution on the same input data.

% color list of edges with given list of colors Cs
color_graph([],_).
color_graph([e(CI,CJ)|Es],Cs):-
take_one_color(CI,Cs,OtherCs),
take_one_color(CJ,OtherCs,_), % CJ is anything except CI
color_graph(Es,Cs).

take_one_color(C,[C|Cs],Cs).
take_one_color(C,[Other|Cs1],[Other|Cs2]):-
take_one_color(C,Cs1,Cs2).

% test data: undirected graph as list of edges
go:-Edges=[
  e(V1,V2),e(V2,V3),e(V1,V3),e(V3,V4),e(V4,V5),
  e(V5,V6),e(V4,V6),e(V2,V5),e(V1,V6)
],
Colors=[r,g,b],
color_graph(Edges,Colors),
  writeln(Edges),
  fail
;   writeln(done).
/*
?- go.
[e(r,g),e(g,b),e(r,b),e(b,r),e(r,b),e(b,g),e(r,g)
],e(g,b),e(r,g)]
[e(r,g),e(g,b),e(r,b),e(b,g),e(g,r),e(r,b),e(g,b)
],e(g,r),e(r,b)]
[e(r,b),e(b,g),e(r,g),e(g,r),e(r,g),e(g,b),e(r,b)
],e(b,g),e(r,b)]
[e(r,b),e(b,g),e(r,g),e(g,b),e(b,r),e(r,g),e(b,g)
],e(b,r),e(r,g)]
[e(g,r),e(r,b),e(g,b),e(b,r),e(r,g),e(g,b),e(r,b)
],e(r,g),e(g,b)]
[e(g,r),e(r,b),e(g,b),e(b,g),e(g,b),e(b,r),e(g,r)
],e(r,b),e(g,r)]
[e(g,r),e(r,b),e(g,b),e(b,g),e(g,b),e(b,r),e(g,r)
],e(b,g),e(g,r)]
[e(b,r),e(r,g),e(b,g),e(g,r),e(r,b),e(b,g),e(r,g)
],e(r,b),e(b,g)]
[e(b,r),e(r,g),e(b,g),e(g,b),e(b,g),e(g,r),e(b,r)
],e(r,g),e(b,r)]
[e(b,g),e(g,r),e(b,r),e(r,g),e(b,r),e(g,b),e(b,g)
],e(g,b),e(b,g)]
[e(b,g),e(g,r),e(b,r),e(r,g),e(b,r),e(r,g),e(b,g)
],e(g,r),e(b,g)]
done
true.
*/