Solutions for CIS301 Exercise set 6
10 points

Q1. p -> q, ~q |- ~p

```
1. p -> q       premise
2. ~q           premise
... 3. p             assumption
... 4. q             ->e 1,3
... 5. _|_           ~e 4,2
6. ~p            ~i 3-5
```

Q2. q -> r, p -> ~r |- p -> ~q

```
1.  q -> r      premise
2.  p -> ~r     premise
... 3.  p           assumption
... 4.  ~r          ->e 2,3
...  ... 5.    q         assumption
...  ... 6.    r         ->e 1,5
...  ... 7. _|_           ~e 6,4
...  8.  ~q         ~i 5-7
9. p -> ~q      ->i 3-8
```

Q3. p -> q, ~q |- p -> r

```
1.  p -> q        premise
2.  ~q            premise
... 3.  p             assumption
... 4.  q             ->e 1,3
... 5.  _|_           ~e 4,2
... 6.  r             _|_e 5
7.  p -> r       ->i 3-6
```

Q1. (p ^ q), r |- (p ^ r) ^ (s v q)

```
1 (p ^ q)            premise
2  r                  premise
3  p                  ^e1 1
4  p ^ r              ^i  3,2
5  q                  ^e2  1
6  s v q              vi2  5
7  (p ^ r) ^ (s v q)  ^i 4,6
```

Q2. p v (q ^ r) |- (p v q) ^ (p v r)

```
1  p v (q ^ r)     premise
+-----------------------
   2  p           assumption
   3  p v q        vi1  1
   4  p v r        vi1  2
   5 (p v q) ^ (p v r)   ^i 3,4
+------------------------
```

```
+-----------------------------
   6  q ^ r          assumption
   7  q              ^e1   6
   8  p v q          vi2 7
   9  r              ^e2  6
  10  p v r          vi2 9
  11  (p v q) ^ (p v r)   ^i 8,10
+----------------------
12 (p v q) ^ (p v r)   ve 1,2-5,6-11
```

Q3. p -> q, (p ^ q) -> r, p |-â r ^ q

```
1  p -> q        premise
2  (p ^ q) -> r  premise
3  p             premise
4  q             ->e 1,3
5  p ^ q         ^i 3,4
6  r             ->e 2,5
7  r ^ q         ^i 6,4
```

Q4. p -> (q -> r) |- (p ^ q) -> r

```
1  p -> (q -> r)    premise
+---------------------------
   2  p ^ q           assumption
   3  p               ^e1 2
   4  q -> r          ->e 1,3
   5  q               ^e2 2
   6  r               ->e 4,5
+---------------------------
7  (p ^ q) -> r     ->i 2-6
```

Q5. p -> r, q -> s |- (p v q) -> (r v s)

```
1  p -> r               premise
2  q -> s               premise
+---------------------------------------
   3  p v q                assumption
   +-----------------------------------
      4   p                    assumption
      5   r                    ->e 1,4
      6   r v s                vi1 5
   +-----------------------------------
   +-----------------------------------
      7  q                    assumption
      8  s                    ->e 2,7
      9  r v s                vi2 8
   +-----------------------------------
   10   r v s              ve 3,4-6,7-9
+---------------------------------------
11  (p v q) -> (r v s)   ->i 3-10
```

```
def eval(t) :

   """pre:  t  is an ETREE,
             where ETREE ::= NUMERAL |
[ OP, ETREE1, ETREE2 ]
                   NUMERAL is a string,
and OP is  "+" or "-"
       post: ans is the numerical
meaning of t
    """
 if isinstance(t,str) and t.isdigit() :
        ans = int(t)
    else :  # t is a list, [op, t1, t2]
        op = t[0]
        t1 = t[1]
        t2 = t[2]
        ans1 = eval(t1)
        # assert:  ans1  is the
numerical meaning of t1
        ans2 = eval(t2)
        # assert:  ans2  is the
numerical meaning of t2
        if op == "+" :
            ans = ans1 + ans2
        elif op == "-" :
            ans = ans1 - ans2
    return ans
```

7,6

... 9. c: int-array
assumption
... 10. 4: int num
... 11. c[4]:int index
9, 10
12. def g(int-array c) return c[4] end:
int-array -> int   def 9-11

13. g(b): int call 12,7
14. a = g(b): int assign 3,
13

Q5. Write a proof that shows this program is well-typed:

```
new int a;
a = 2;
new int[9] b;
b[a + a] = a;
def g(int-array c) return c[4] end;
a = g(b)
```

1. new int a: int decvar

2. 2: int num
3. a: int ref 1
4. a = 2: int assign
3,2

5. new int[9] b: int-array
decarray

6. a + a: int add
3,3
7. b: int-array
arrayref 5
8. b[a+a]: int index