1. No, this is an opinion. If there is no definite true/false answer, it is not a proposition.
2. Hikaru is shorter than Yutaka
3. (a)$3 > x \; or \; x \geq 4$

   (b) Some people weigh less than 100 pounds.

4. This statement is true because the premise of the statement is false. Conditionals with a false premise are true
5. If it is pleasant to for you to travel in economy class, then you are 6 feet tall or shorter.
6. Only if: If we could not visit the stars, then we did not have an FTL drive.
   Sufficient: If we had an FTL drive, then we could visit the stars.
   Necessary: If we do not have an FTL drive, we can not visit the stars.
   Contrapositive: If we can not visit the stars, we do not have an FTL drive.
   Unless: We could visit the stars unless we did not have an FTL drive.
7. $\exists x \forall y (x \leq y)$
8. (p →q) → (¬p →¬q) Given
   $(\neg p \lor q) \to (p \lor \neg q)$ By Equivalency
   $\neg(\neg p \lor q) \lor (p \lor \neg q)$ By Equivalency
   $(p \land \neg q) \lor (p \lor \neg q)$ By De Morgan
   $\big((p \land \neg q) \lor p\big) \lor \big((p \land \neg q) \lor \neg q\big)$ By Distribution
   $p \lor \neg q$ By Absorption
   $q \to p$ By Equivalency

9.

(a)

Consider all cases of A_k, B_k:

A_k	B_k	C_k
0	0	0
0	1	1
1	0	1
1	1	0

In the case drive A failed, $A_{new} = B_k$ XOR C_k

B_k	C_k	A_{new}
0	0	0
1	1	0
0	1	1
1	0	1

A_{new} is the same as the original A_k.

Additionally, this also works to recover drive B: $B_{new} = A_k$ XOR C_k

(b)

No, using AND would require both inputs to be high for C_k to be used to recover the information.

B_k	C_k	A_{new}
0	0	0
1	0	0
0	0	0
1	1	1

C_k values were changed to match the original values of A_k. $A_{new} \neq A_k$ following the rule, $A_{new} = B_k$ AND $C_{k.}$

There are fewer high values saved in A_{new}

(c)

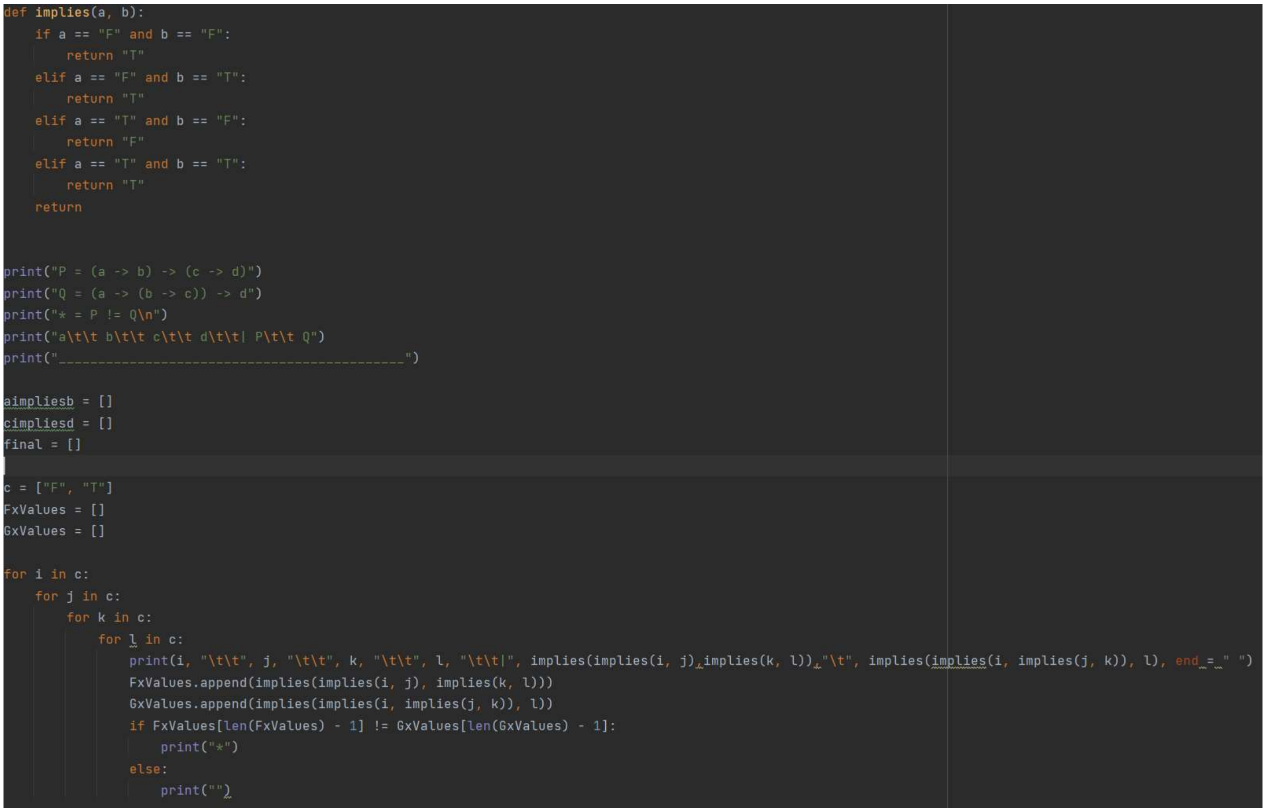
No, using OR would allow for high value to be saved to C_k more often. Using the same example as in B following the rules $C_k = A_k$ OR B_k, and $A_{new} = C_k$ OR B_k

B_k	C_k	A_{new}
0	0	0
1	1	1
0	1	1
1	1	1

There are more high values saved in A_{new}.

10. True, there exists x = 0 which makes any value of y in the domain of discourse equal 0.
11. No, say P(x) = x > 5 and Q(x) = x ≤ 5. If P(4) ∨ Q(6), both predicates are false. Whereas, ∀x(P (x) ∨Q(x)), would be true.
12. For every x, there is a unique y and unique z such that x is friends with y and z.
13. False, it could be possible that x does not have any friends, or has one friend.

14.



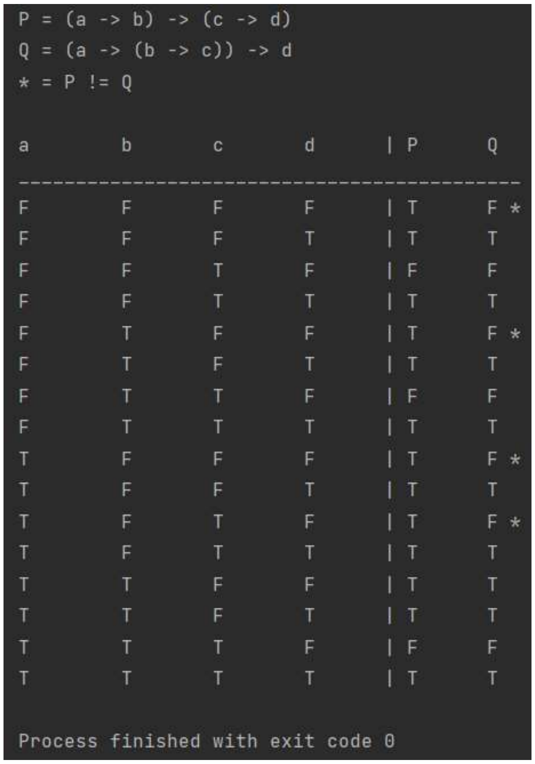
```python
def implies(a, b):
    if a == "F" and b == "F":
        return "T"
    elif a == "F" and b == "T":
        return "T"
    elif a == "T" and b == "F":
        return "F"
    elif a == "T" and b == "T":
        return "T"
    return


print("P = (a -> b) -> (c -> d)")
print("Q = (a -> (b -> c)) -> d")
print("* = P != Q\n")
print("a\t\t b\t\t c\t\t d\t\t| P\t\t Q")
print("_____________________________________________")

aimpliesb = []
cimpliesd = []
final = []

c = ["F", "T"]
FxValues = []
GxValues = []

for i in c:
    for j in c:
        for k in c:
            for l in c:
                print(i, "\t\t", j, "\t\t", k, "\t\t", l, "\t\t|", implies(implies(i, j),implies(k, l)),"\t", implies(implies(i, implies(j, k)), l), end = " ")
                FxValues.append(implies(implies(i, j), implies(k, l)))
                GxValues.append(implies(implies(i, implies(j, k)), l))
                if FxValues[len(FxValues) - 1] != GxValues[len(GxValues) - 1]:
                    print("*")
                else:
                    print("")
```

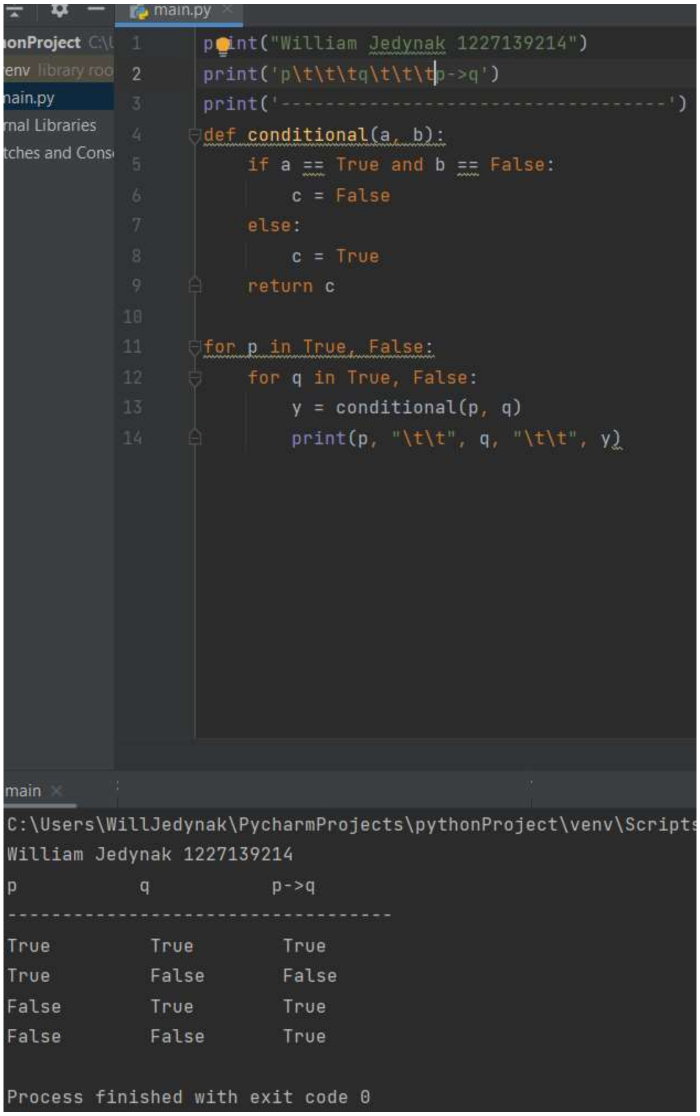


```
P = (a -> b) -> (c -> d)
Q = (a -> (b -> c)) -> d
* = P != Q

a        b        c        d        | P      Q
_____________________________________________
F        F        F        F        | T      F *
F        F        F        T        | T      T
F        F        T        F        | F      F
F        F        T        T        | T      T
F        T        F        F        | T      F *
F        T        F        T        | T      T
F        T        T        F        | F      F
F        T        T        T        | T      T
T        F        F        F        | T      F *
T        F        F        T        | T      T
T        F        T        F        | T      F *
T        F        T        T        | T      T
T        T        F        F        | T      T
T        T        F        T        | T      T
T        T        T        F        | F      F
T        T        T        T        | T      T

Process finished with exit code 0
```

EXTRA CREDIT:



```python
print("William Jedynak 1227139214")
print('p\t\t\tq\t\t\tp->q')
print('----------------------------------')
def conditional(a, b):
    if a == True and b == False:
        c = False
    else:
        c = True
    return c

for p in True, False:
    for q in True, False:
        y = conditional(p, q)
        print(p, "\t\t", q, "\t\t", y)
```

```
C:\Users\WillJedynak\PycharmProjects\pythonProject\venv\Scripts
William Jedynak 1227139214
p           q           p->q
----------------------------------
True         True         True
True         False        False
False        True         True
False        False        True

Process finished with exit code 0
```