__/__/____

Arfath Kabir (1BM18CS130)

2)

```
def tell (kb, rule)
    kb.append (rule)

combinations = [(True, True, True), (True, True, False),
                (True, False, True), (True, False, False)
                (False, True, True), (False, True, False)
                (False, False, True), (False, False, False)]

def ask (kb, q):
    for c in combinations:
    s = all (rule (c) for rule in kb)
    f = q (c)
    print (s, f)
    if s != f and s != False:
        return 'Does not entail'
    return 'Entails'

kb = []
r1 = lambda x : x[0] or [1] and (x[0] and x[1])
tell (kb, r1)

r2 = lambda x : (x[0] or x[1]) and x[2]
tell (kb, r2)

q = lambda x : x[0] and x[1] and (x[0] or x[1])
ask (kb, q)

r3 = lambda x : (x[0] and x[1]) or x[2]
r4 = lambda x : not x[1] or not x[2]
r5 = lambda x : not x[0] and x[1]
```

__/__/____

```
tell (kb, r3)
tell (kb, r4)
tell (kb, r5)

q = lambda x : x[0] and not x[1]
ask (kb, q)
```