7.Implement unification in first order logic.

```
def unify(s1, s2, theta={}):

    if theta is None:
        return None

    if s1 == s2:
        return theta

    if isinstance(s1, str) and s1.islower():
        return unify_var(s1, s2, theta)

    if isinstance(s2, str) and s2.islower():
        return unify_var(s2, s1, theta)

    if isinstance(s1, tuple) and isinstance(s2, tuple) and len(s1) == len(s2):
        return unify(s1[1:], s2[1:], unify(s1[0], s2[0], theta))

    return None

def unify_var(var, x, theta):
    if var in theta:
        return unify(theta[var], x, theta)
    elif x in theta:
        return unify(var, theta[x], theta)
    elif occurs_check(var, x, theta):
        return None
    else:
        theta[var] = x
        return theta

def occurs_check(var, x, theta):
    if var == x:
        return True
    elif isinstance(x, str) and x.islower() and x in theta:
        return occurs_check(var, theta[x], theta)
    elif isinstance(x, tuple):
        for arg in x:
            if occurs_check(var, arg, theta):
                return True
    return False
```

```
s1 = ('p', 'x', ('f', 'x'), ('y'))
s2 = ('p', 'a', 'y', ('f', 'x'))

substitution = unify(s1, s2)

if substitution:
    print("Unification successful:")
    print(f"Substitution: {substitution}")
else:
    print("Unification failed.")
```

Output:

```
Unification successful:
Substitution: {'x': 'a', 'y': ('f', 'x')}
```

Observation