Implement Unification Algorithm in First Order Logic

```
def unify(s1, s2):
    subst = {}
    return unifyRecursive(s1, s2, subst)

def unifyRecursive(s1, s2, subst):
    s1 = substitute(s1, subst)
    s2 = substitute(s2, subst)

    if s1 == s2:
        return subst

    if isinstance(s1, list) and isinstance(s2, list):
        if len(s1) != len(s2):
            return None
        for i in range(len(s1)):
            result = unifyRecursive(s1[i], s2[i], subst)
            if result is None:
                return None
            subst = result
        return subst

    if isinstance(s1, str) and s1.islower():
        return unifyVariable(s1, s2, subst)

    if isinstance(s2, str) and s2.islower():
        return unifyVariable(s2, s1, subst)

    return None

def unifyVariable(var, x, subst):
    if var in subst:
        return unifyRecursive(subst[var], x, subst)
    if x == var:
        return subst
    if isinstance(x, list) and any(v == var for v in x):
        return None
    if isinstance(x, str) and x.islower() and x in subst:
        return unifyRecursive(var, subst[x], subst)

    newSubst = subst.copy()
    newSubst[var] = x
    return newSubst

def substitute(expr, subst):
    if isinstance(expr, list):
        return [substitute(e, subst) for e in expr]
    elif isinstance(expr, str) and expr.islower() and expr in subst:
        return substitute(subst[expr], subst)
    else:
        return expr

def get_input(prompt):
    while True:
        user_input = input(prompt)
        terms = user_input.split()

        if all(term.islower() or term.isalpha() for term in terms):
            return terms
        else:
            print("Invalid input. Please enter space-separated terms (e.g., 'P x y'). Try again.")

if __name__ == "__main__":
    print("Unification Algorithm in First-Order Logic")
    print("Output: 1BM22CS290")
    s1 = get_input("Enter the first expression (e.g., 'P x y'): ")
    s2 = get_input("Enter the second expression (e.g., 'P a y'): ")

    result = unify(s1, s2)

    if result:
        print("Unification successful!")
        print("Substitution:", result)
    else:
```

```
    print("The expressions cannot be unified.")
```

```
Unification Algorithm in First-Order Logic
Output: 1BM22CS290
Enter the first expression (e.g., 'P x y'): F x y
Enter the second expression (e.g., 'P a y'): F a b
Unification successful!
Substitution: {'x': 'a', 'y': 'b'}
```