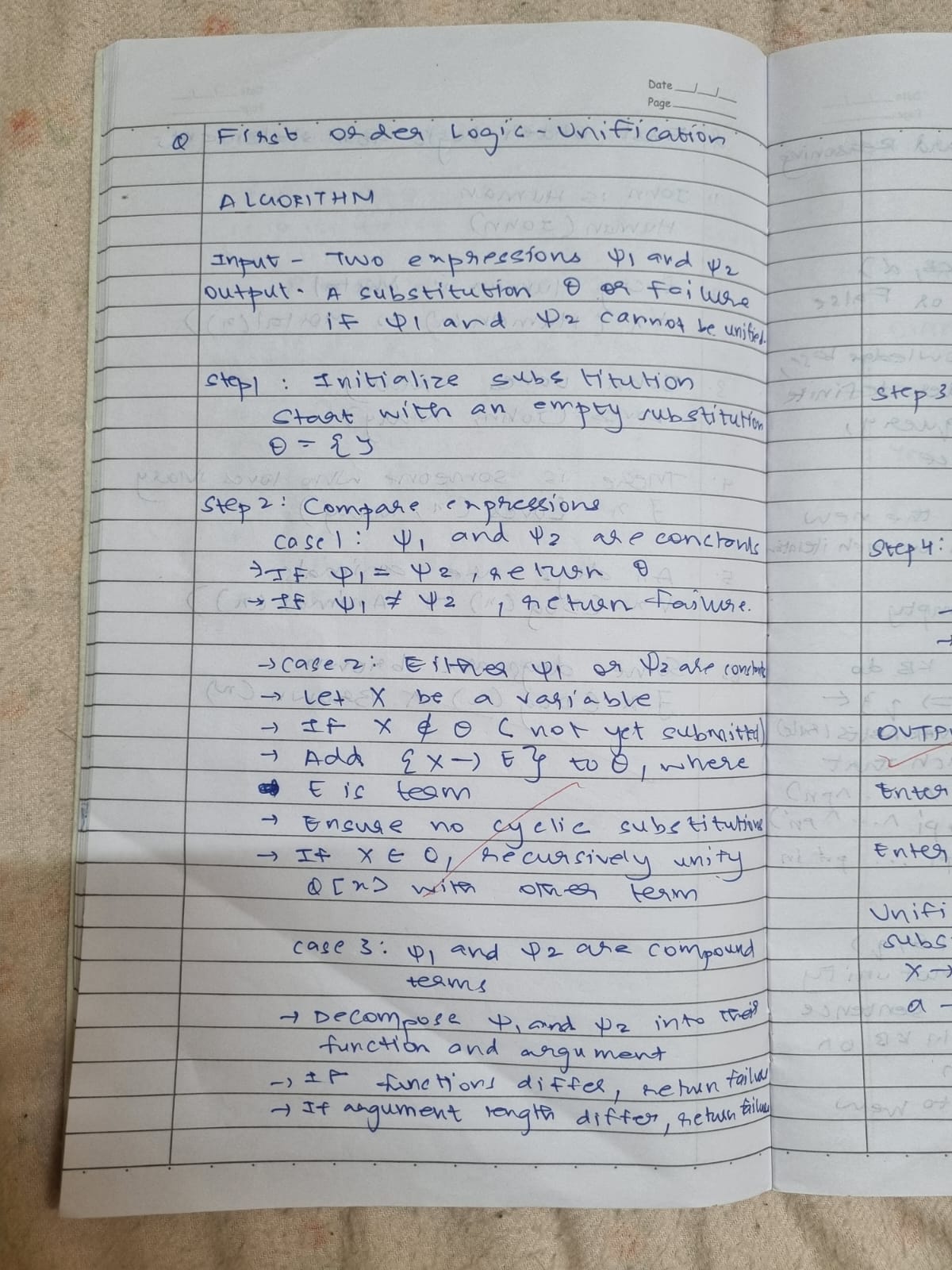
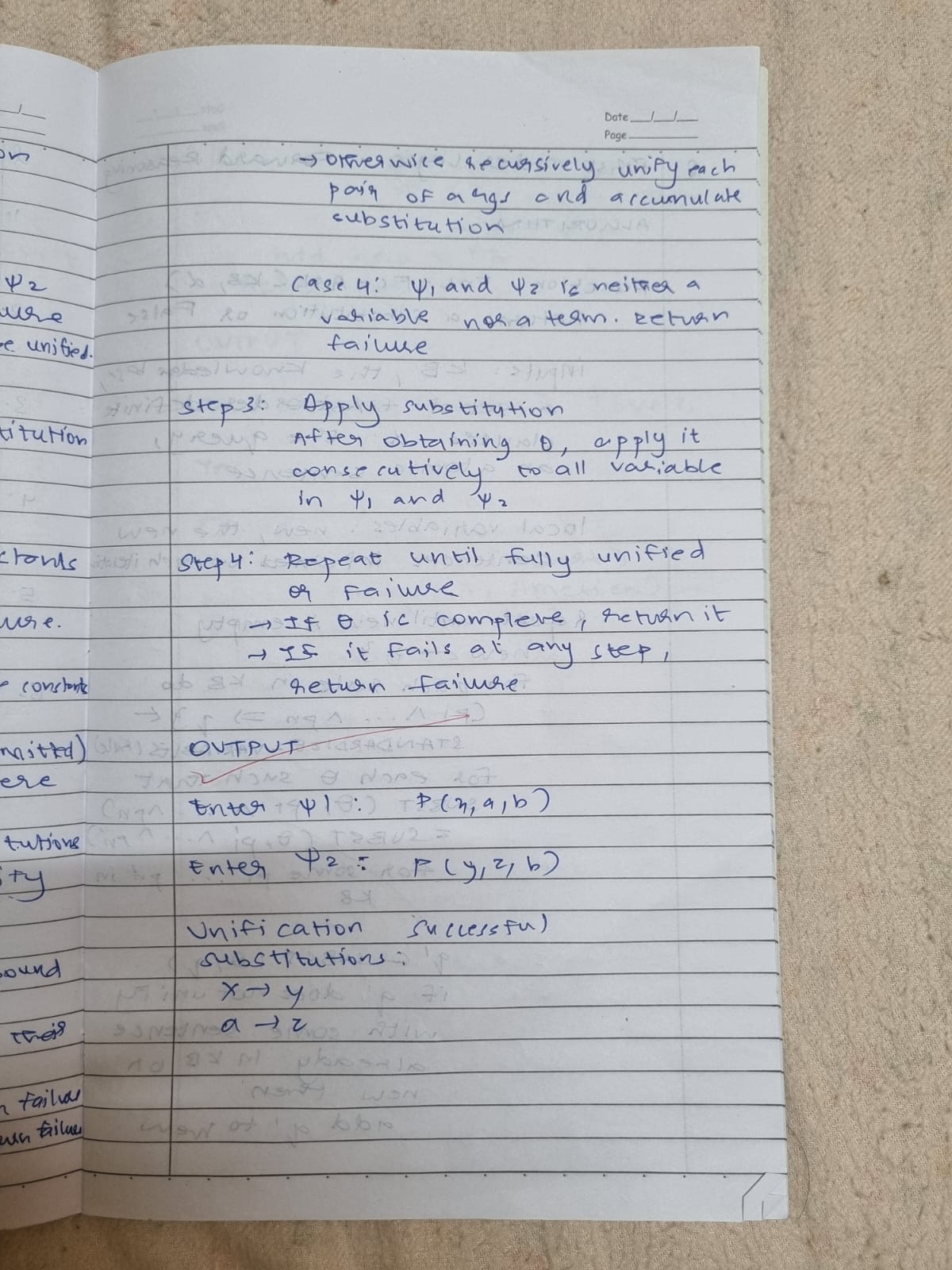
**First Order Logic - Unification**

**Algorithm:**

****

****

**Code:**

def is\_variable(term):

    return isinstance(term, str) and term.islower() and len(term) == 1

def occurs\_check(var, term):

    if var == term:

        return True

    if isinstance(term, tuple):

        return any(occurs\_check(var, sub\_term) for sub\_term in term[1:])

    return False

def unify(psi1, psi2, subst=None):

    if subst is None:

        subst = {}

    if psi1 == psi2:

        return subst

    if is\_variable(psi1):

        return unify\_variable(psi1, psi2, subst)

    if is\_variable(psi2):

        return unify\_variable(psi2, psi1, subst)

    if isinstance(psi1, tuple) and isinstance(psi2, tuple):

        if psi1[0] != psi2[0]:

            return None

        if len(psi1) != len(psi2):

            return None

        for arg1, arg2 in zip(psi1[1:], psi2[1:]):

            subst = unify(arg1, arg2, subst)

            if subst is None:  # Step 5(b)

                return None

        return subst

  return None

def unify\_variable(var, term, subst):

    if var in subst:

        return unify(subst[var], term, subst)

    elif isinstance(term, str) and term in subst:

        return unify(var, subst[term], subst)

    elif occurs\_check(var, term):

        return None

    else:

        subst[var] = term

        return subst

def parse\_term(term\_str):

    term\_str = term\_str.strip()

    if "(" in term\_str and term\_str.endswith(")"):

        predicate = term\_str[:term\_str.index("(")]

        arguments = term\_str[term\_str.index("(") + 1:-1].split(",")

        return (predicate,) + tuple(arg.strip() for arg in arguments)

    return term\_str

psi1 = input("Enter Ψ1: ").strip()

psi2 = input("Enter Ψ2: ").strip()

term1 = parse\_term(psi1)

term2 = parse\_term(psi2)

result = unify(term1, term2)

if result is not None:

    print("Unification Successful!")

    print("Substitutions:")

    for var, value in result.items():

        print(f"{var} -> {value}")

else:

    print("Unification Failed!")

**Output:**

