Sagar Bangari(1BM22CS231)

LAB – 7 Unification in First Order Logic

```
import ast
def unify(x, y, subst=None):
  if subst is None:
    subst = {}

  if x == y:
    return subst
  elif isinstance(x, str) and x.islower():
    return unify_var(x, y, subst)
  elif isinstance(y, str) and y.islower():
    return unify_var(y, x, subst)
  elif isinstance(x, list) and isinstance(y, list):
    if len(x) != len(y):
      return "FAILURE"
    if x[0] != y[0]:
      return "FAILURE"
    for xi, yi in zip(x[1:], y[1:]):
      subst = unify(xi, yi, subst)
      if subst == "FAILURE":
        return "FAILURE"
    return subst
  else:
    return "FAILURE"
```

```
def unify_var(var, x, subst):
    if var in subst:
        return unify(subst[var], x, subst)
    elif isinstance(x, (list, tuple)) and tuple(x) in subst:
        return unify(var, subst[tuple(x)], subst)
    elif occurs_check(var, x):
        return "FAILURE"
    else:
        subst[var] = x
        return subst

def occurs_check(var, x):
    if var == x:
        return True
    elif isinstance(x, list):
        return any(occurs_check(var, xi) for xi in x)
    return False

def unify_and_check(expr1, expr2):
    result = unify(expr1, expr2)
    if result == "FAILURE":
        return False, None
    return True, result

def display_result(expr1, expr2, is_unified, subst):
    if not is_unified:
```

```
        print("Result: Unification Failed")
    else:
        print("Result: Unification Successful")
        print("Substitutions:", {k: v for k, v in subst.items()})

def get_expression_input(prompt):
    while True:
        try:
            user_input = input(prompt)
            expr = ast.literal_eval(user_input)
            if isinstance(expr, list):
                return expr
            else:
                print("Please enter a valid list expression (e.g., ['p', 'x', ['F', 'y']]).")
        except (ValueError, SyntaxError):
            print("Invalid input. Please provide a valid list expression (e.g., ['p', 'x', ['F', 'y']]).")

def main():
    print("Enter the first expression to unify (in list format):")
    expr1 = get_expression_input("Expression 1: ")
    print("Enter the second expression to unify (in list format):")
    expr2 = get_expression_input("Expression 2: ")
    is_unified, result = unify_and_check(expr1, expr2)
    display_result(expr1, expr2, is_unified, result)
if __name__ == "__main__":
    main()
```

Output:

```
Enter the first expression to unify (in list format):
Expression 1: ['p', 'x']
Enter the second expression to unify (in list format):
Expression 2: ['p', 'x', ['F', 'y']]
Result: Unification Failed
```

```
Enter the first expression to unify (in list format):
Expression 1: ['p', 'x', ['F', 'y']]
Enter the second expression to unify (in list format):
Expression 2: ['p', 'a', ['F', ['g','x']]]
Result: Unification Successful
Substitutions: {'x': 'a', 'y': ['g', 'x']}
```