LAB 9:

Program: Create knowledge base consisting of FOL statements and prove given query using resoltuion.

Code:

```
def resolve_clause(c1, c2):

resolvent = set(c1)

for literal in c2:

if f"¬{literal}" in resolvent:

resolvent.remove(f"¬{literal}")

return resolvent

return None

def resolution(kb, query):

negated_query = {f"¬{literal}" for literal in query}

kb = kb.copy()

kb.append(frozenset(negated_query))

clauses = [frozenset(clause) for clause in kb]

while True:

new_clauses = set()

for i in range(len(clauses)):

for j in range(i + 1, len(clauses)):

resolvent = resolve_clause(clauses[i], clauses[j])

if resolvent:

new_clauses.add(frozenset(resolvent))

if frozenset() in new_clauses:
```

```
return "Yes"
if new_clauses.issubset(clauses):
return "No"
clauses.extend(new_clauses)
kb = [
{"Food(Banana)", "Enjoys(Ravi, Banana)"},
{"Food(Pizza)", "Enjoys(Ravi, Pizza)"},
{"Eats(Sam, Idli)", "Alive(Sam)", "Food(Idli)"},
{"Eats(Sam, f)", "Alive(Sam)", "Food(f)"},
{"Eats(Sam, f)", "Eats(Bill, f)"},
]
query = {"Enjoys(Ravi, Idli)"}
result = resolution(kb, query)
print("Result:", result)
```