```python
def negate(lit):
    return lit[1:] if lit.startswith("~") else "~" + lit

def resolve(c1, c2):
    for l1 in c1:
        for l2 in c2:
            if l1 == negate(l2):
                new_clause = list(set(c1 + c2))
                new_clause.remove(l1)
                new_clause.remove(l2)
                return [new_clause]
    return []

def resolution(kb, query):
    clauses = kb + [[negate(q)] for q in query]
    new = []

    while True:
        pairs = [(clauses[i], clauses[j]) for i in range(len(clauses)) for j in range(i+1, len(clauses))]
        for (c1, c2) in pairs:
            res = resolve(c1, c2)
            if res == [[]]:
                print("the query is not satisfiable")
                return
            for r in res:
                if r not in clauses and r not in new:
                    new.append(r)
        if not new:
            print("the query is satisfiable")
            return
        clauses += new

# Input
print("Knowledge base:")
print("[")
kb = [["~P", "Q"], ["P"], ["~Q", "R"], ["~R"]]
for clause in kb:
    print(clause)
print("]")

query = ["R"]

print("Query:")
print(query)

print("Knowledge Base:", kb)
print("Query:", query)

resolution(kb, query)
```

Output

```
Knowledge base:
[
['~P', 'Q']
['P']
['~Q', 'R']
['~R']
]
Query:
['R']
Knowledge Base: [['~P', 'Q'], ['P'], ['~Q', 'R'], ['~R']]
Query: ['R']
the query is not satisfiable

=== Code Execution Successful ===
```