```
def negate(literal):
    return literal[1:] if literal.startswith("~") else "~" + literal
def resolve(ci, cj):
    for lit in ci:
        if negate(lit) in cj:
            new_clause = list(set(ci + cj))
            new_clause.remove(lit)
            new_clause.remove(negate(lit))
            return [new_clause]
    return []
def resolution(kb, query):
    clauses = kb + [[negate(q)] for q in query]
    while True:
        new = []
        for i in range(len(clauses)):
            for j in range(i + 1, len(clauses)):
                resolvents = resolve(clauses[i], clauses[j])
                if [] in resolvents:
                    return True
                for r in resolvents:
                    if r not in new:
                        new.append(r)
        if all(n in clauses for n in new):
            return False
        clauses += new
kb = [["~P", "Q"], ["P"], ["~Q", "R"], ["~R"]]
query = ["R"]
print("Knowledge Base:", kb)
print("Query:", query)
print("The query is satisfiable." if resolution(kb, query) else "The query is not satisfiable.")
```

```
Knowledge Base: [['~P', 'Q'], ['P'], ['~Q', 'R'], ['~R']]
Query: ['R']
The query is satisfiable.

[Program finished]
```