Lab program - 6
Knowledge base entailment

Algorithm

Step 1 : Tokenization:

Tokenize both 'sentence 1' and 'sentence 2' into individual words, considering punctuation and whitespaces.

Step 2: Normalization:

Convert all words to lowercase to make the comparison case-insensitive.

Step 3: comparison:

Initialize a boolean variable, entailment Holds to true.

Step: End:

End of algorithm.

Code.

```
from sympy.logic.boolalg import Implies, NOT

from sympy.abc import p, q
```

```python
class PropositionalKnowledge:
    def __init__(self):
        self.knowledge_base = set()

    def add_statement(self, statement):
        return query.simplify() in self.knowledge_base
```

```python
kb = PropositionKnowledgeBase()

kb.add_statement(Implies(p, q))
kb.add_statement(Not(q))


query1 = p
query2 = Not(p)

result1 = kb.check_entailment(query1)
result2 = kb.check_entailment(query2)

print(f"Does '{query1}' logically follow from
    the knowledge base? {result1}")

print(f"Does '{query2}' logically follow from
    the knowledge base? {result2}")
```

Output
___

Does 'p' logically follow from the knowledge
base? True.

Does '~p' logically follow from the
knowledge base? False.