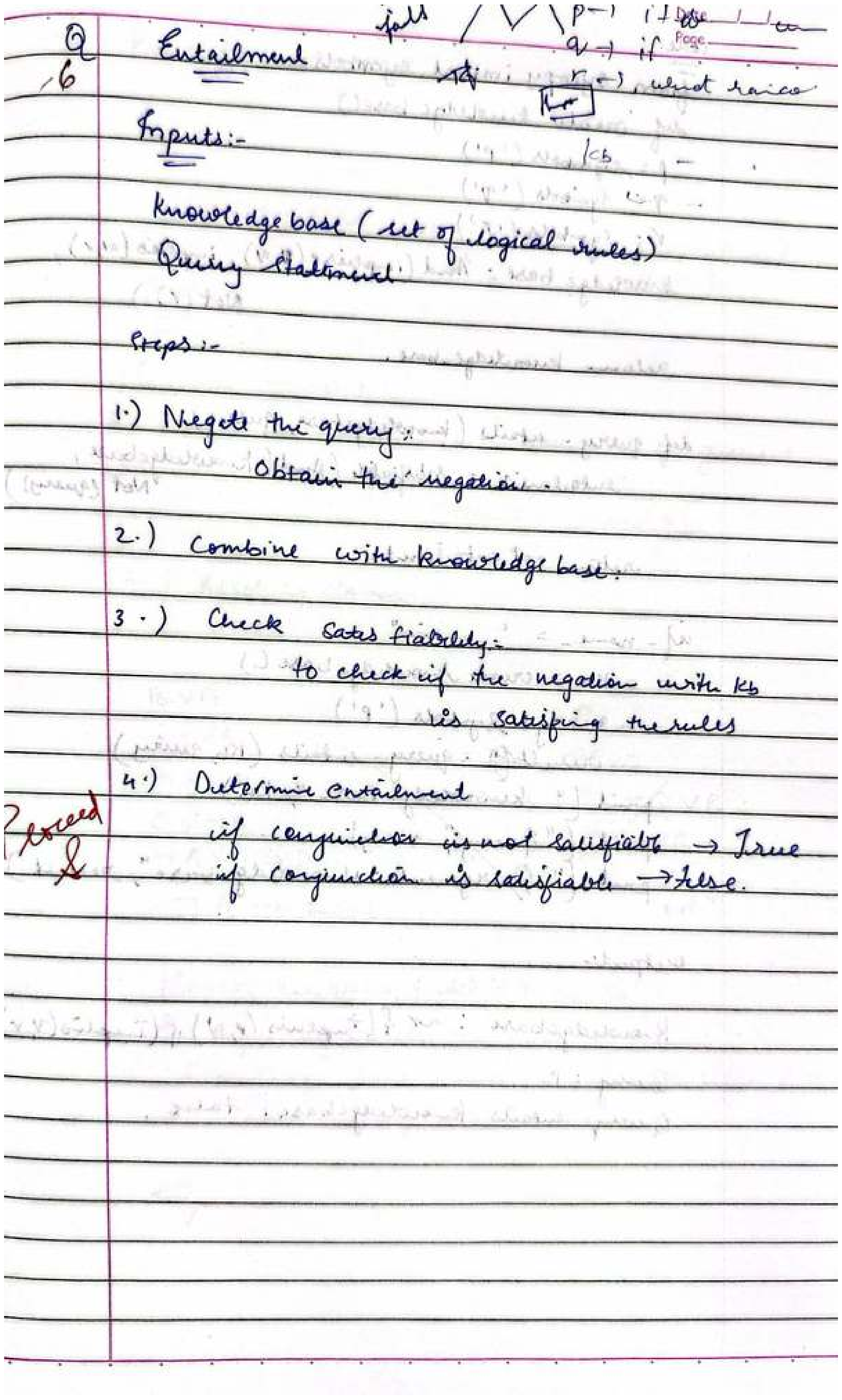
Knowledge-based entailment:-
Code:-



fall / V \ p→ if
q→ if
r→ what rains
[illegible] kb

Q 6 Entailment

Inputs:-

knowledge base (set of logical rules)
Query statement

Steps:-

1.) Negate the query:
Obtain the negation.

2.) Combine with knowledge base.

3.) Check satisfiability:
to check if the negation with kb is satisfying the rules

4.) Determine entailment
if conjunction is not satisfiable → True
if conjunction is satisfiable → False.

Proceed
S

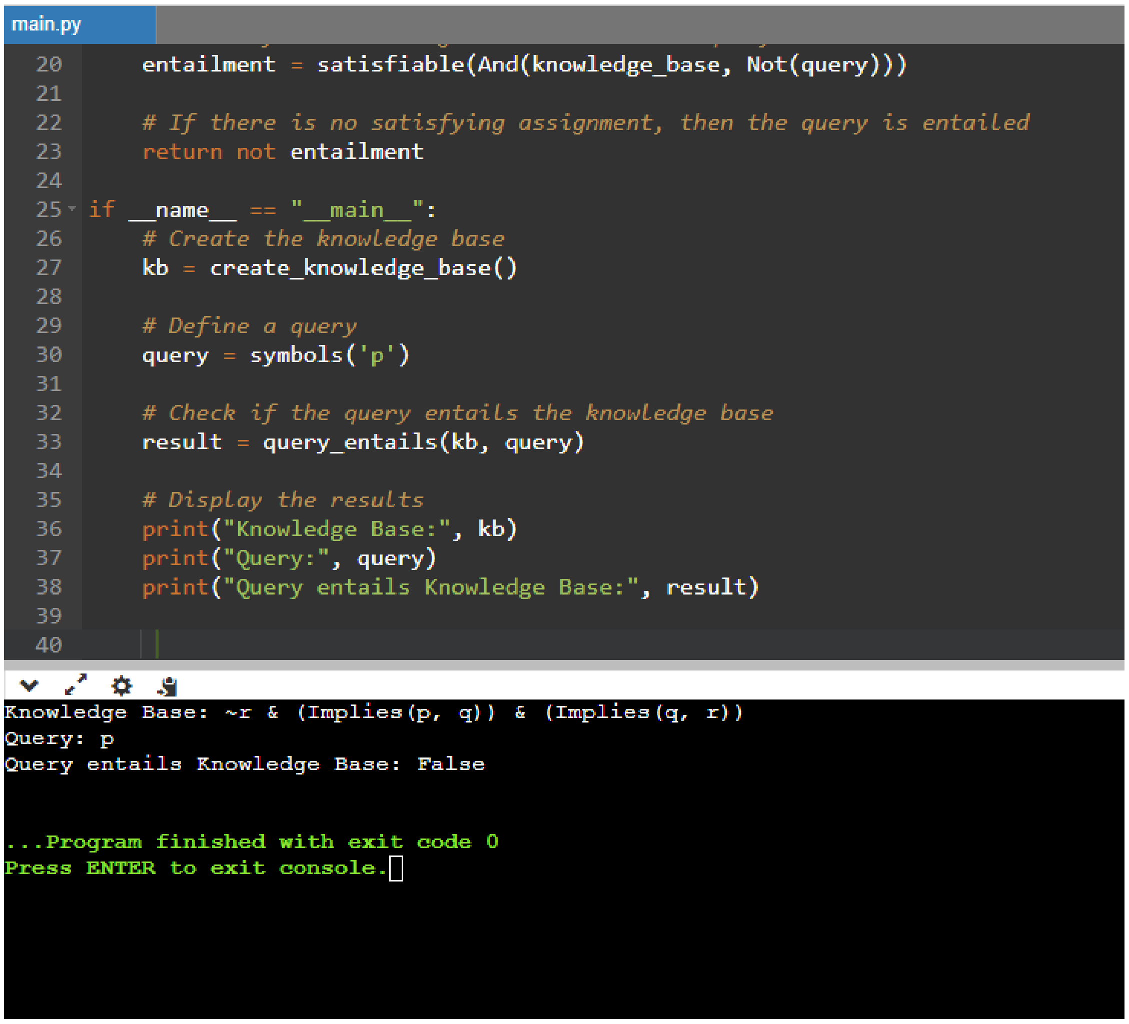
code:-

```
from sympy import symbols
def create knowledge base()
 p = symbols ('p')
 q = symbols ('q')
 r = symbols ('r')
 knowledge base = And (impliese(p,q), implies (q,r),
                                          Not (r) )

 return knowledge_base.

def query - entails (knowledgebase, query)
      entailment = satisfiable (And ( knowledgebase,
                                                   Not (query)

      return not entailment.

if _ name _ = '__main__"
      kb = create knowledge base ()
       query = Symbols ('p')
       result = query - entails (kb, query)
   print (" knowledge Base ", kb)
   print (" Query", query)
   print (" Query entails knowledge base", result)
```

Output:-

```
Knowledgebase : ~r & (Implies (p, q) & (Implies (q, r
Query : p
Query entails knowledgebase , False .
```

output:-


main.py

```python
    entailment = satisfiable(And(knowledge_base, Not(query)))

    # If there is no satisfying assignment, then the query is entailed
    return not entailment

if __name__ == "__main__":
    # Create the knowledge base
    kb = create_knowledge_base()

    # Define a query
    query = symbols('p')

    # Check if the query entails the knowledge base
    result = query_entails(kb, query)

    # Display the results
    print("Knowledge Base:", kb)
    print("Query:", query)
    print("Query entails Knowledge Base:", result)

```

```
Knowledge Base: ~r & (Implies(p, q)) & (Implies(q, r))
Query: p
Query entails Knowledge Base: False


...Program finished with exit code 0
Press ENTER to exit console.
```