Homework:

(a) let $X := (((P \Rightarrow Q) \vee S) \Leftrightarrow T)$

$P \in \mathcal{P}(v)$

$Q \in \mathcal{P}(v)$

$(P \Rightarrow Q) \in \mathcal{P}(v)$

$S \in \mathcal{P}(v)$

$((P \Rightarrow Q) \vee S) \in \mathcal{P}(v)$

$T \in \mathcal{P}(v)$

$(((P \Rightarrow Q) \vee S) \Rightarrow T) \in \mathcal{P}(v)$

X is a well formed propositional formula (wff)

(b) let $x := ((P \Rightarrow (Q \wedge (S \Rightarrow T))))$

$S \in P(\pi)$

$T \in P(\pi)$

$(S \Rightarrow T) \in P(\pi)$

$Q \in P(\pi)$

$(Q \wedge (S \Rightarrow T)) \in P(\pi)$

$P \in P(\pi)$

$(P \Rightarrow (Q \wedge (S \Rightarrow T))) \in P(\pi)$

$((P \Rightarrow (Q \wedge (S \Rightarrow T)))) \notin P(\pi)$

X is NOT a well formed propositional formula

(c) let $X := (\neg(B(\neg Q)) \wedge R)$

$Q \in P(\pi)$

$(\neg Q) \in P(\pi)$

$(B(\neg Q)) \notin P(\pi)$

$(\neg(B(\neg Q)) \wedge R) \notin P(\pi)$

X is NOT a well formed propositional formula

(d) let $X := (P \wedge ((\neg Q) \wedge (\neg(\neg(Q \Leftrightarrow (\neg R))))))$

$R \in \mathbb{P}(V)$

$(\neg R) \in \mathbb{P}(V)$

$Q \in \mathbb{P}(V)$

$(Q \Leftrightarrow (\neg R)) \in \mathbb{P}(V)$

$(\neg(Q \Leftrightarrow (\neg R))) \in \mathbb{P}(V)$

$(\neg(\neg(Q \Leftrightarrow (\neg R)))) \in \mathbb{P}(V)$

$(\neg Q) \in \mathbb{P}(V)$

$((\neg Q) \wedge (\neg(\neg(Q \Leftrightarrow (\neg R)))))$

$P \in \mathbb{P}(V)$

$(P \wedge ((\neg Q) \wedge (\neg(\neg(Q \Leftrightarrow (\neg R)))))) \in \mathbb{P}(V)$

X is a well formed propositional formula

(e) let $X := ((P \vee Q) \Rightarrow \neg(P \vee Q)) \wedge (P \vee (\neg(\neg Q)))$

$P \in \mathbb{P}(V)$

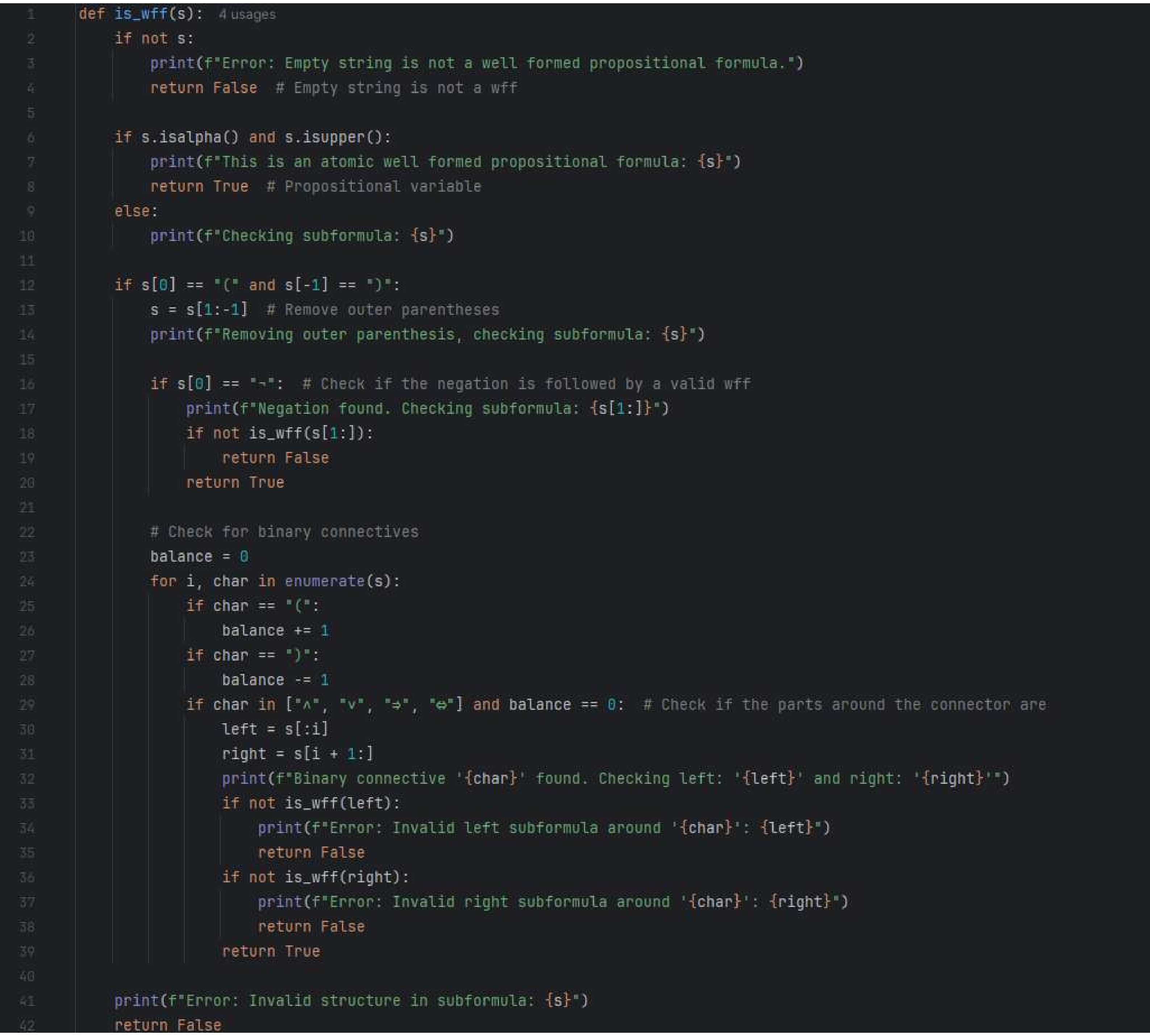
$Q \in \mathbb{P}(V)$

$(P \vee Q) \in \mathbb{P}(V)$

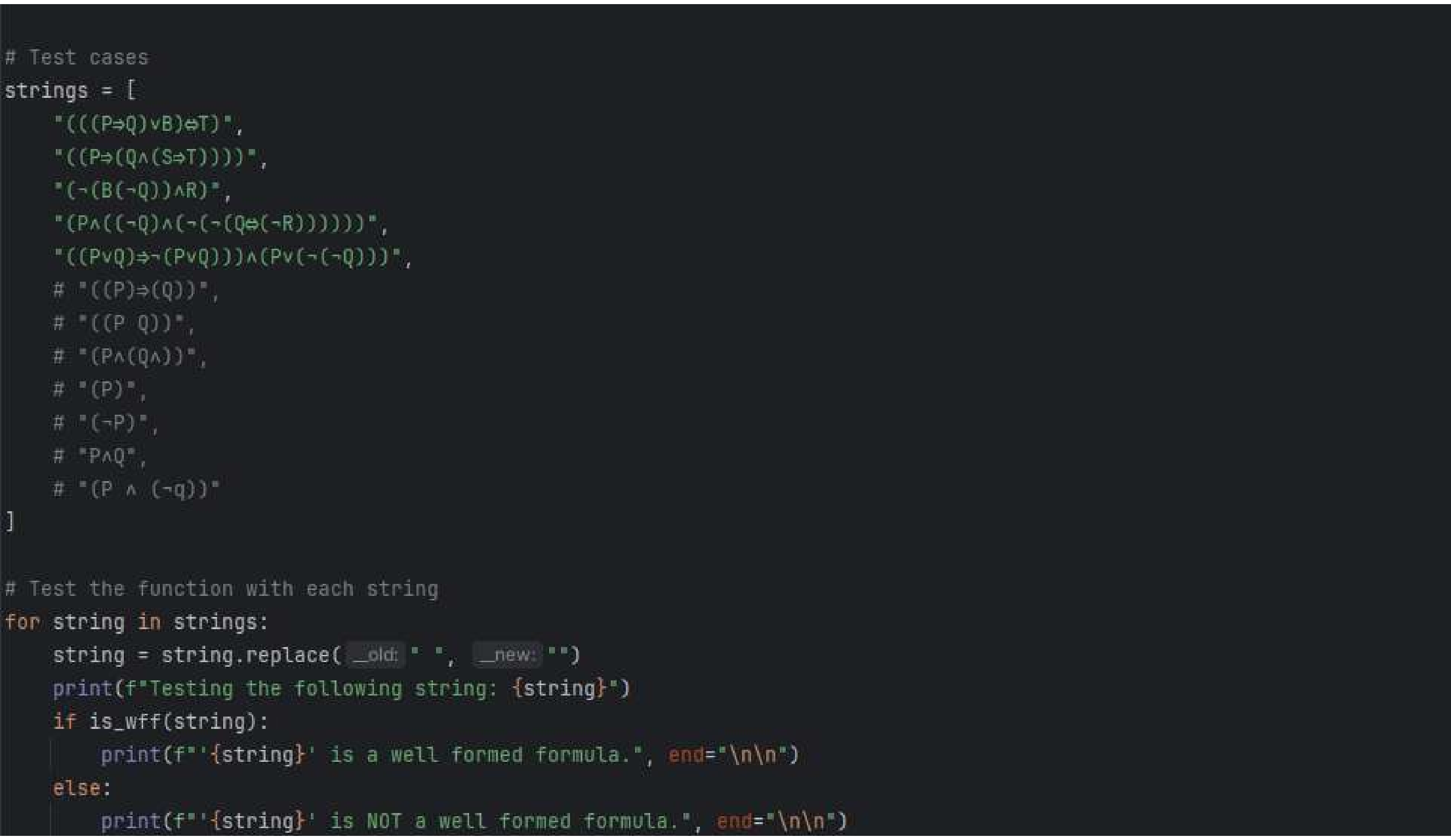
$\neg(P \vee Q) \notin \mathbb{P}(V)$

$((P \vee Q) \Rightarrow \neg(P \vee Q)) \wedge (P \vee (\neg(\neg Q))) \notin \mathbb{P}(V)$

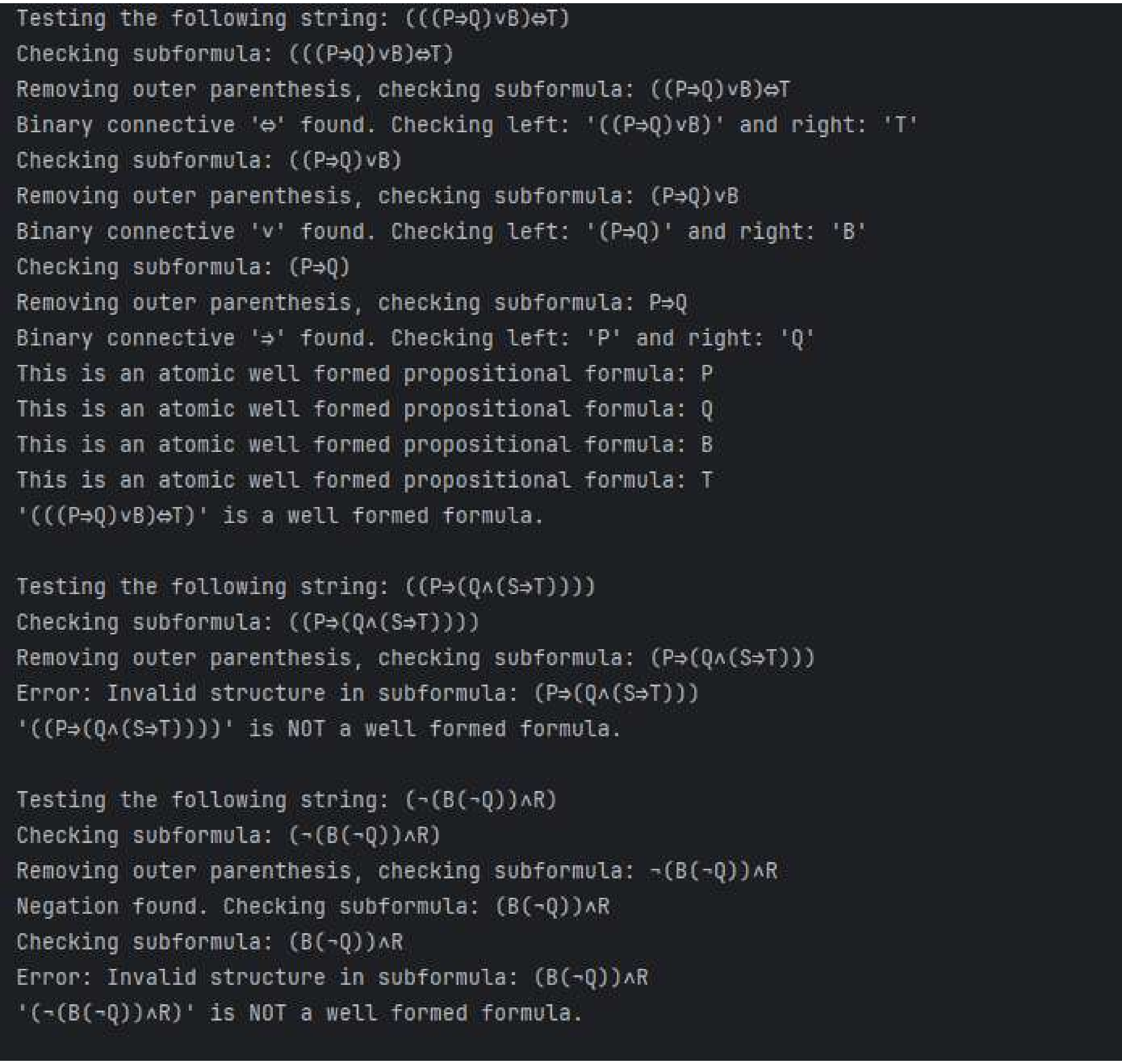
X is NOT a well formed propositional formula



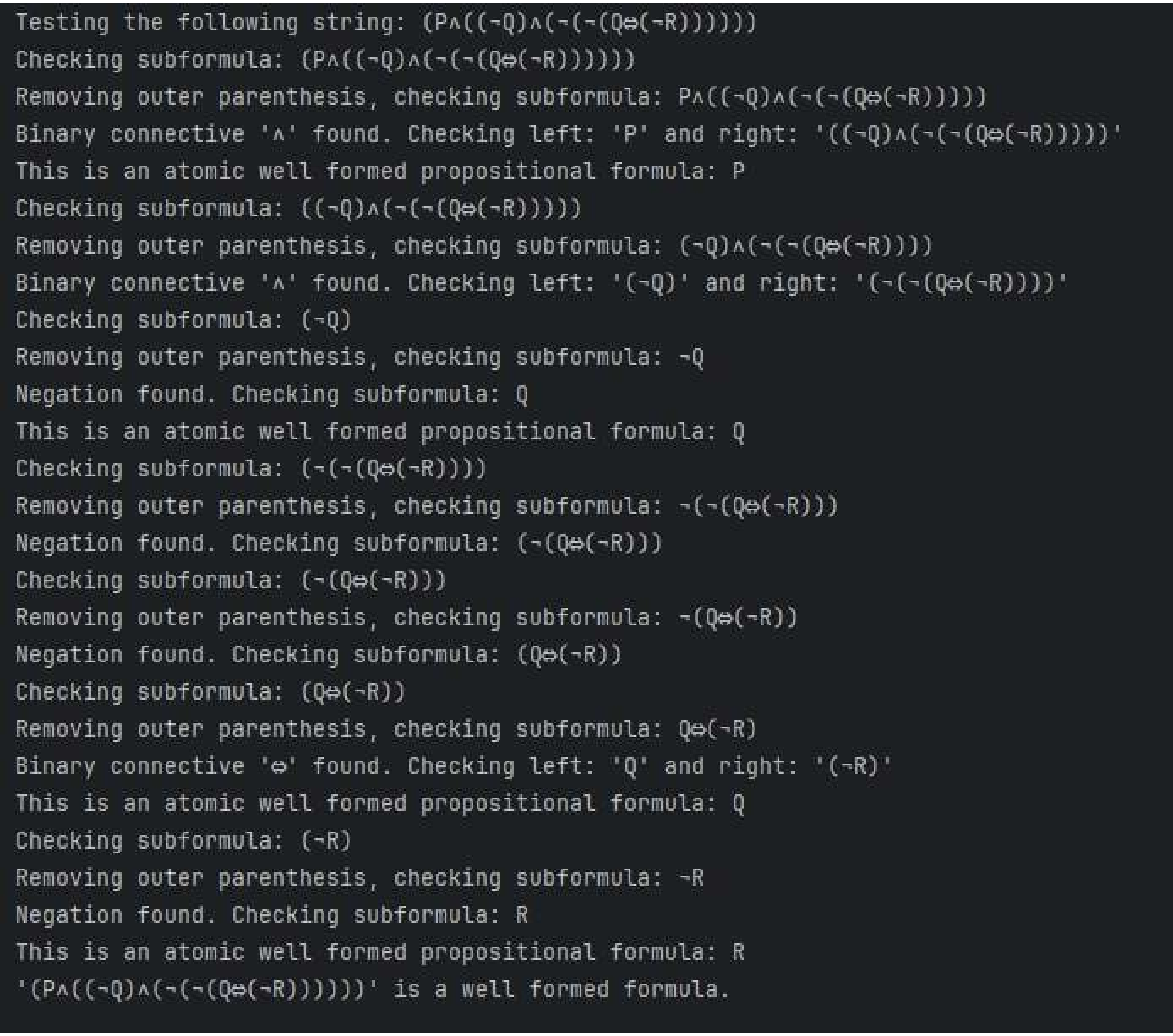
```python
def is_wff(s):
    if not s:
        print(f"Error: Empty string is not a well formed propositional formula.")
        return False  # Empty string is not a wff

    if s.isalpha() and s.isupper():
        print(f"This is an atomic well formed propositional formula: {s}")
        return True  # Propositional variable
    else:
        print(f"Checking subformula: {s}")

    if s[0] == "(" and s[-1] == ")":
        s = s[1:-1]  # Remove outer parentheses
        print(f"Removing outer parenthesis, checking subformula: {s}")

        if s[0] == "¬":  # Check if the negation is followed by a valid wff
            print(f"Negation found. Checking subformula: {s[1:]}")
            if not is_wff(s[1:]):
                return False
            return True

        # Check for binary connectives
        balance = 0
        for i, char in enumerate(s):
            if char == "(":
                balance += 1
            if char == ")":
                balance -= 1
            if char in ["∧", "∨", "⇒", "⇔"] and balance == 0:  # Check if the parts around the connector are
                left = s[:i]
                right = s[i + 1:]
                print(f"Binary connective '{char}' found. Checking left: '{left}' and right: '{right}'")
                if not is_wff(left):
                    print(f"Error: Invalid left subformula around '{char}': {left}")
                    return False
                if not is_wff(right):
                    print(f"Error: Invalid right subformula around '{char}': {right}")
                    return False
                return True

    print(f"Error: Invalid structure in subformula: {s}")
    return False
```



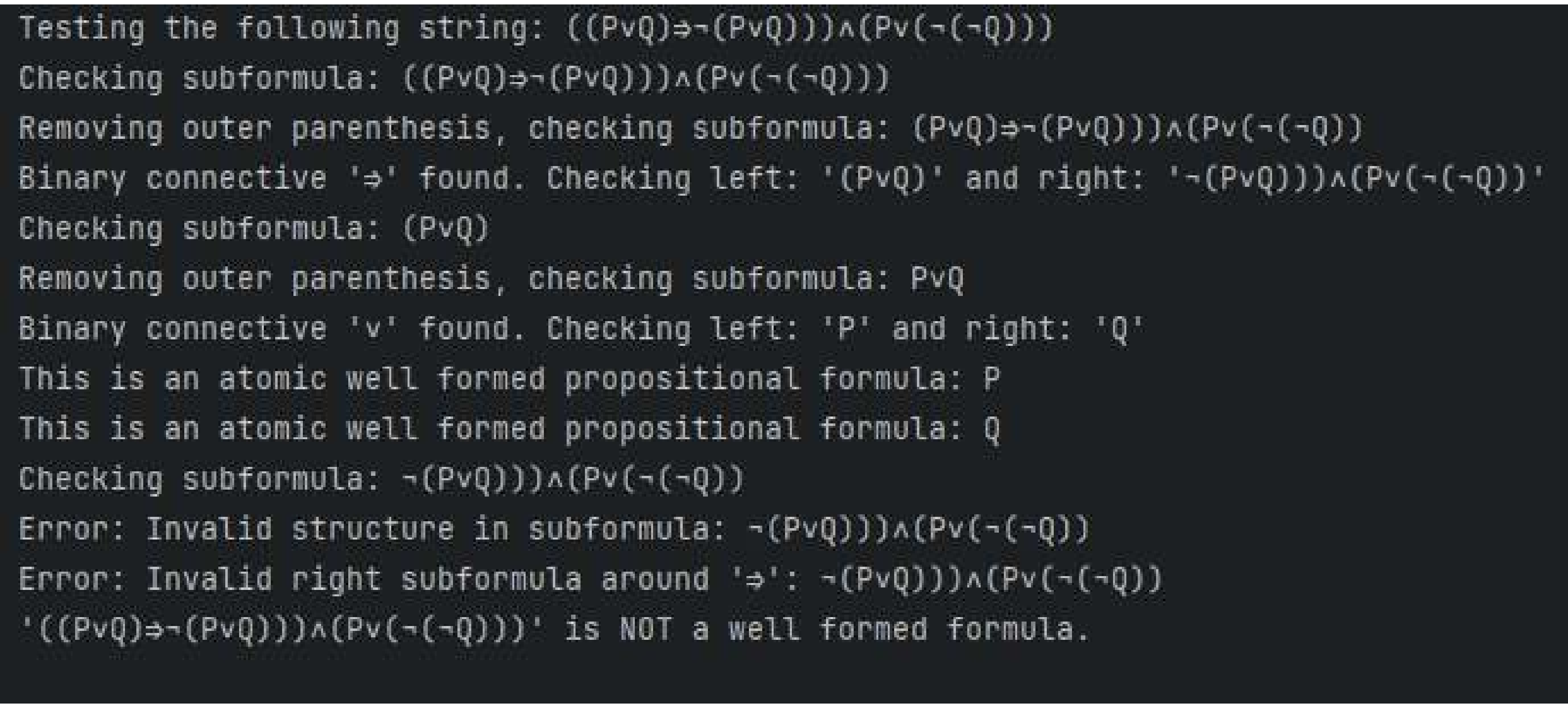
```python
# Test cases
strings = [
    "(((P⇒Q)∨B)⇔T)",
    "((P⇒(Q∧(S⇒T))))",
    "(¬(B(¬Q))∧R)",
    "(P∧((¬Q)∧(¬(¬(Q⇔(¬R)))))))",
    "((P∨Q)⇒¬(P∨Q)))∧(P∨(¬(¬Q)))",
    # "((P)⇒(Q))",
    # "((P Q))",
    # "(P∧(Q∧))",
    # "(P)",
    # "(¬P)",
    # "P∧Q",
    # "(P ∧ (¬q))"
]

# Test the function with each string
for string in strings:
    string = string.replace( __old: " ",  __new: "")
    print(f"Testing the following string: {string}")
    if is_wff(string):
        print(f"'{string}' is a well formed formula.", end="\n\n")
    else:
        print(f"'{string}' is NOT a well formed formula.", end="\n\n")
```



```
Testing the following string: (((P⇒Q)∨B)⇔T)
Checking subformula: (((P⇒Q)∨B)⇔T)
Removing outer parenthesis, checking subformula: ((P⇒Q)∨B)⇔T
Binary connective '⇔' found. Checking left: '((P⇒Q)∨B)' and right: 'T'
Checking subformula: ((P⇒Q)∨B)
Removing outer parenthesis, checking subformula: (P⇒Q)∨B
Binary connective '∨' found. Checking left: '(P⇒Q)' and right: 'B'
Checking subformula: (P⇒Q)
Removing outer parenthesis, checking subformula: P⇒Q
Binary connective '⇒' found. Checking left: 'P' and right: 'Q'
This is an atomic well formed propositional formula: P
This is an atomic well formed propositional formula: Q
This is an atomic well formed propositional formula: B
This is an atomic well formed propositional formula: T
'(((P⇒Q)∨B)⇔T)' is a well formed formula.

Testing the following string: ((P⇒(Q∧(S⇒T))))
Checking subformula: ((P⇒(Q∧(S⇒T))))
Removing outer parenthesis, checking subformula: (P⇒(Q∧(S⇒T)))
Error: Invalid structure in subformula: (P⇒(Q∧(S⇒T)))
'((P⇒(Q∧(S⇒T))))' is NOT a well formed formula.

Testing the following string: (¬(B(¬Q))∧R)
Checking subformula: (¬(B(¬Q))∧R)
Removing outer parenthesis, checking subformula: ¬(B(¬Q))∧R
Negation found. Checking subformula: (B(¬Q))∧R
Checking subformula: (B(¬Q))∧R
Error: Invalid structure in subformula: (B(¬Q))∧R
'(¬(B(¬Q))∧R)' is NOT a well formed formula.
```



```
Testing the following string: (P∧((¬Q)∧(¬(¬(Q⇔(¬R))))))
Checking subformula: (P∧((¬Q)∧(¬(¬(Q⇔(¬R))))))
Removing outer parenthesis, checking subformula: P∧((¬Q)∧(¬(¬(Q⇔(¬R)))))
Binary connective '∧' found. Checking left: 'P' and right: '((¬Q)∧(¬(¬(Q⇔(¬R)))))'
This is an atomic well formed propositional formula: P
Checking subformula: ((¬Q)∧(¬(¬(Q⇔(¬R)))))
Removing outer parenthesis, checking subformula: (¬Q)∧(¬(¬(Q⇔(¬R))))
Binary connective '∧' found. Checking left: '(¬Q)' and right: '(¬(¬(Q⇔(¬R))))'
Checking subformula: (¬Q)
Removing outer parenthesis, checking subformula: ¬Q
Negation found. Checking subformula: Q
This is an atomic well formed propositional formula: Q
Checking subformula: (¬(¬(Q⇔(¬R))))
Removing outer parenthesis, checking subformula: ¬(¬(Q⇔(¬R)))
Negation found. Checking subformula: (¬(Q⇔(¬R)))
Checking subformula: (¬(Q⇔(¬R)))
Removing outer parenthesis, checking subformula: ¬(Q⇔(¬R))
Negation found. Checking subformula: (Q⇔(¬R))
Checking subformula: (Q⇔(¬R))
Removing outer parenthesis, checking subformula: Q⇔(¬R)
Binary connective '⇔' found. Checking left: 'Q' and right: '(¬R)'
This is an atomic well formed propositional formula: Q
Checking subformula: (¬R)
Removing outer parenthesis, checking subformula: ¬R
Negation found. Checking subformula: R
This is an atomic well formed propositional formula: R
'(P∧((¬Q)∧(¬(¬(Q⇔(¬R))))))' is a well formed formula.
```



```
Testing the following string: ((P∨Q)⇒¬(P∨Q)))∧(P∨(¬(¬Q)))
Checking subformula: ((P∨Q)⇒¬(P∨Q)))∧(P∨(¬(¬Q)))
Removing outer parenthesis, checking subformula: (P∨Q)⇒¬(P∨Q)))∧(P∨(¬(¬Q))
Binary connective '⇒' found. Checking left: '(P∨Q)' and right: '¬(P∨Q)))∧(P∨(¬(¬Q))'
Checking subformula: (P∨Q)
Removing outer parenthesis, checking subformula: P∨Q
Binary connective '∨' found. Checking left: 'P' and right: 'Q'
This is an atomic well formed propositional formula: P
This is an atomic well formed propositional formula: Q
Checking subformula: ¬(P∨Q)))∧(P∨(¬(¬Q))
Error: Invalid structure in subformula: ¬(P∨Q)))∧(P∨(¬(¬Q))
Error: Invalid right subformula around '⇒': ¬(P∨Q)))∧(P∨(¬(¬Q))
'((P∨Q)⇒¬(P∨Q)))∧(P∨(¬(¬Q)))' is NOT a well formed formula.
```