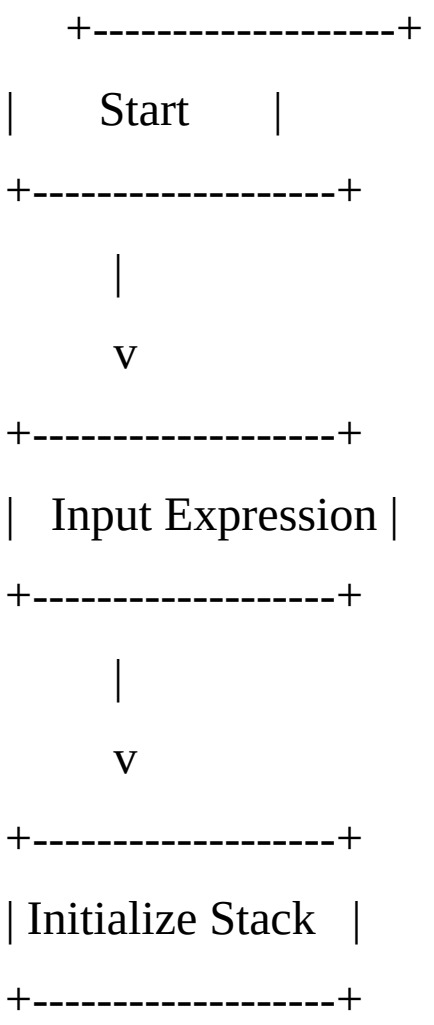
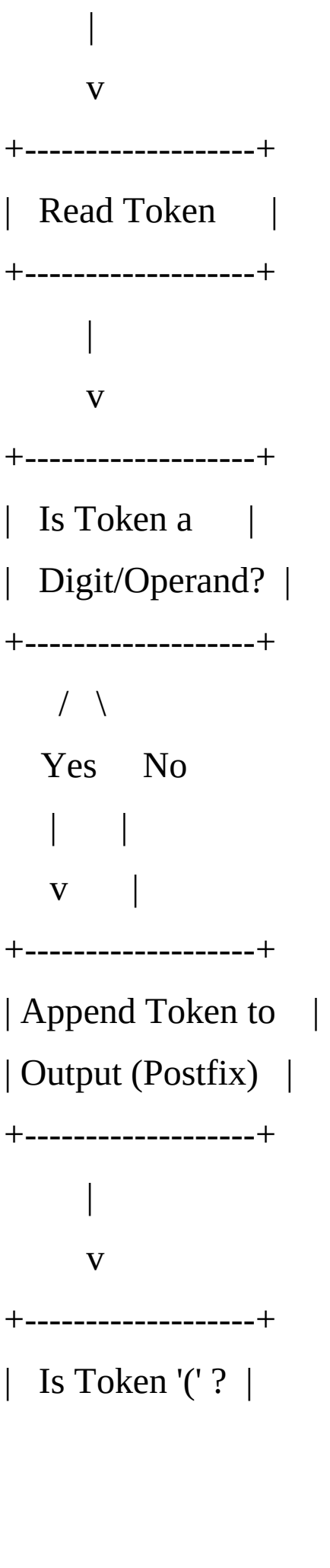
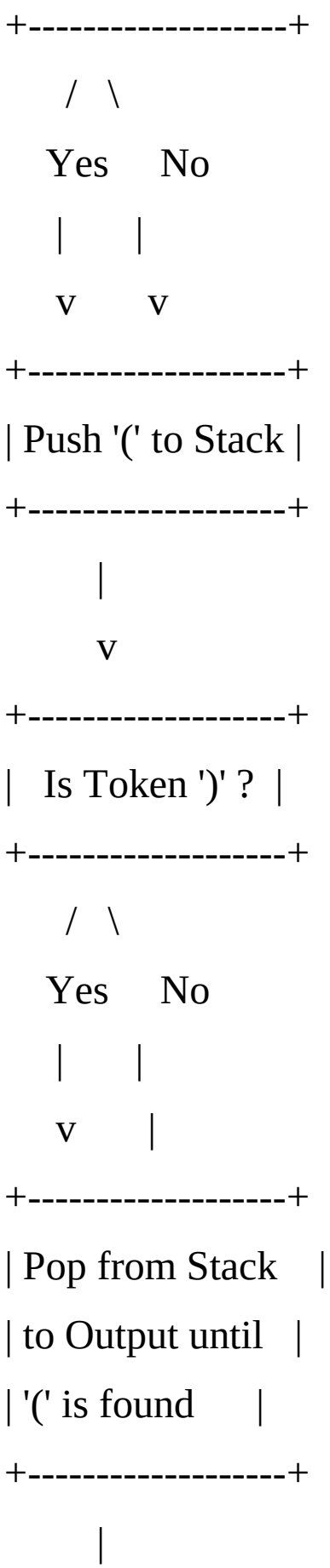
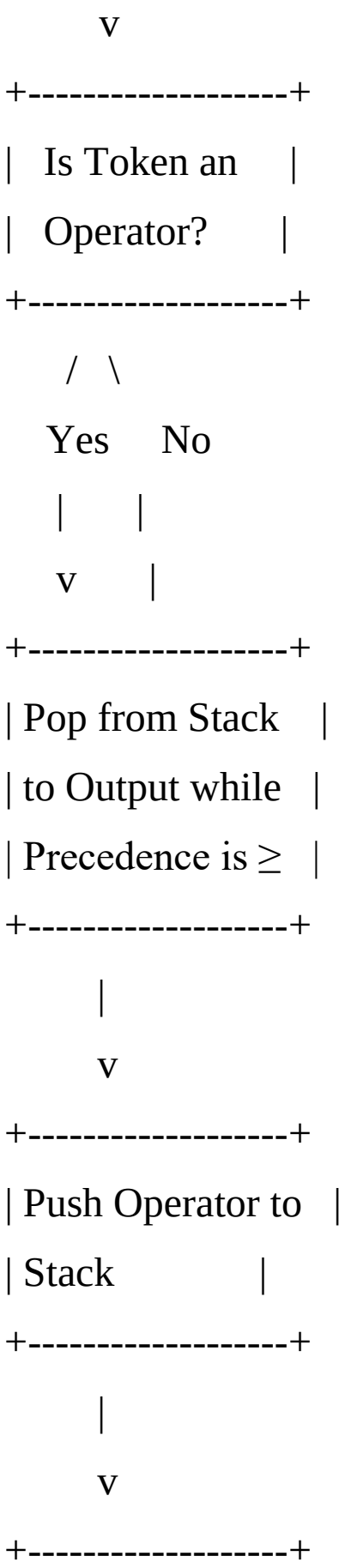
❖ Infix to Postfix:

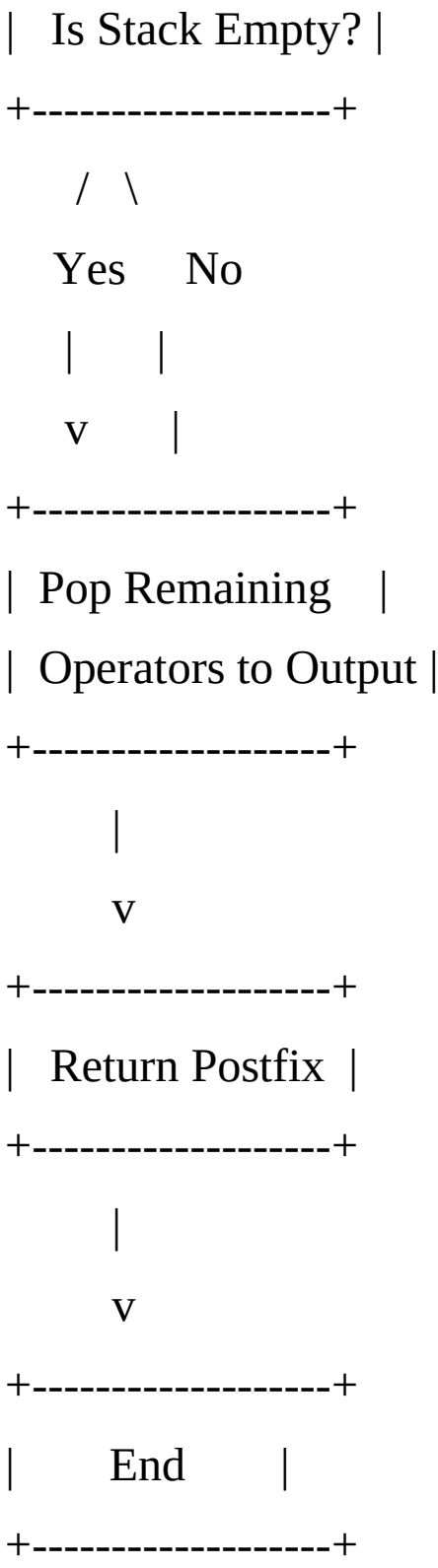
- Explanation- The problem is to convert an infix expression (where operators are placed between operands) into a postfix expression (where operators follow their operands). This conversion is essential for evaluating expressions without the need for parentheses, making it easier for computers to process. The program should read an infix expression from the user and output the equivalent postfix notation.
- Time Complexity- O(n)
- Space Complexity- O(n)
- Flowchart-




Read Token
Is Token a Digit/Operand?
Yes
No
Append Token to Output (Postfix)
Is Token '(' ?


/ \
Yes No
Push '(' to Stack
Is Token ')' ?
/ \
Yes No
Pop from Stack
to Output until
'(' is found


Is Token an Operator?
Yes
No
Pop from Stack to Output while Precedence is ≥
Push Operator to Stack


| Is Stack Empty? |
Yes
No
| Pop Remaining |
| Operators to Output |
| Return Postfix |
| End |