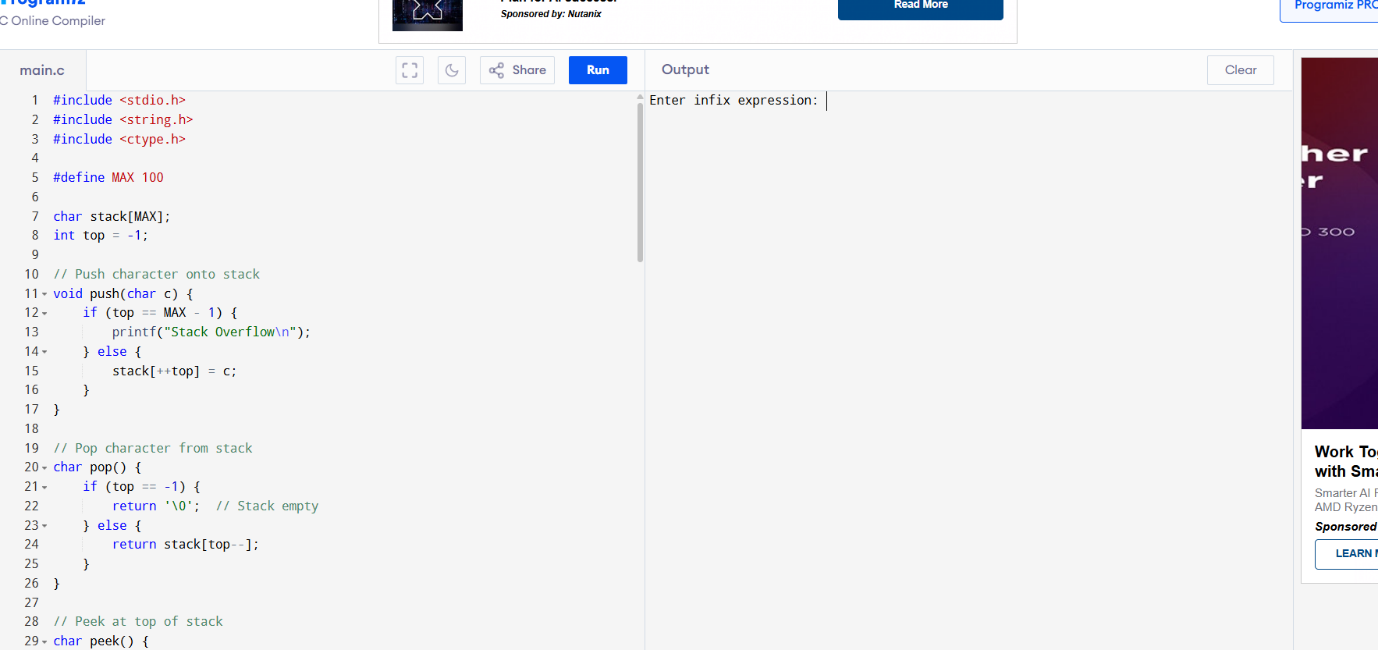
**Aim**

To convert a given infix expression (like A+B\*C) into its postfix equivalent (like ABC\*+) using stack data structure.

**Algorithm (Infix to Postfix)**

1. **Initialize** an empty stack for operators.
2. **Scan** the infix expression from left to right.
3. For each character:
   * If it is an **operand** (number or letter), add it to the postfix expression.
   * If it is an **'('**, push it onto the stack.
   * If it is an **')'**, pop and output from the stack until '(' is encountered. Remove '(' from stack.
   * If it is an **operator** (+, -, \*, /, ^):
     + While the stack is not empty and the operator at the top of the stack has **higher or equal precedence**, pop it and add it to the postfix expression.
     + Push the current operator onto the stack.
4. After the expression is scanned, pop all remaining operators from the stack and add to postfix expression.



**Sample Run**

Enter infix expression: A+B\*C

Postfix expression: ABC\*+