Algorithm for Postfix to Prefix:

1. Scan the Postfix expression from left to right.
2. If the symbol is an operand, then push it onto the Stack
3. If the symbol is an operator, then
    a. pop two operands from the Stack in the following order:
       operand2 = Pop()
       operand1 = Pop()
    b. Create a string by concatenating the two operands and the operator before them.
       string = operator + operand1 + operand2
    c. push the resultant string back to Stack
4. Repeat the above steps until end of Postfix expression.
5. Pop the string representing the Prefix expression on stack and return.

Example:

ab*c+ → +*abc

Operand a -> Push "a"

Operand b -> Push "b"

Operator * ->

    Pop op2, op2 = "b"

    Pop op1, op1="a"

    Create a string, "*ab"

    Push "*ab"

Operand c -> Push "c"

Operator + ->

    Pop op2, op2 = "c"

Pop op1, op1="*ab"

Create a string, "+*abc"

Push "+*abc"

End of the Postfix expression-> Pop the result, "+*abc"

Algorithm for Prefix to Postfix:

1. Scan the Prefix expression in reverse order (from right to left)
2. If the symbol is an operand, then push it onto the Stack
3. If the symbol is an operator, then
    a. pop two operands from the Stack in the following order:
       operand1 = Pop()
       operand2 = Pop()
    b. Create a string by concatenating the two operands and the operator after them.
       string = operand1 + operand2 + operator
    c. Push the resultant string back to Stack
4. Repeat the above steps until end of Prefix expression.
5. Pop the string representing the Postfix expression on stack and return.