

The following shows how a postfix expression is evaluated when using a stack.

Postfix Expression: **7 2 3 * 5 + 8 4 2 / - * -** [Infix Expression: $7 - (2 * 3 + 5) * (8 - 4 / 2)$]

- (1) As you read the expression from left to right, push each operand on the stack (i.e., 12, 5, 30, etc.) until you encounter an operator (i.e., +, -, *, /).
- (2) When you encounter an operator, **POP** the stack twice and perform the appropriate calculation using the operator and the two operands popped. **PUSH** this result back on to the stack.
- (3) Continue to **PUSH** and **POP** until there is no more data. **POP** the stack one more time to obtain the final answer, or value of the expression.



Note: If during the program you can't pop the stack or there is data remaining on the stack at the end, then the postfix expression is wrong.

