**PostfixEvaluation**

**Q:** Given a postfix expression, evaluate it. Postfix expressions are of the form ABop where,

A = number

B = number

op = operator

**Test Cases:**

**Input:** “832+-9\*3/”

**Output:** 9

**Input:** “123\*+4-”

**Output:** 3

**HINT:**

* Use stack instead of converting it to infix and evaluating.
* Convert the character read from the string to the int before pushing into the stack

(Use char-'0' for converting)

**Algorithm:**

Step-1: Create a stack to store all the numbers.

Step-2: Read the character one by one from the given expression.

Step-3: If the character is the digit than push it onto the stack.

--> Before pushing it to the stack convert it to int

Step-4: If the character is the operator than do the following:

--> Pop two numbers say temp1 and temp2

--> Evaluate temp2 Op temp1

--> Push the result back to the stack

Step-5: Goto step-2 if there are still more characters to evaluate, otherwise the only number in the stack is the answer.

**Example:**

Input: 832+-9\*3/

Step-1: Push 8, 3, 2 one by one to the stack by converting it to the number.

Step-2: Now + will come so pop two numbers from the stack and evaluate 2+3 and push 5 (2+3) back to the stack.

Step-3: Now - will come so pop 5 and 8 and push 3 (8-5) to the stack.

Step-4: Push 9 to the stack.

Step-5: \* is read, so pop 3 and 9 and push 27 (9\*3) to the stack.

Step-6: Push 3 to the stack.

Step-7: / is evaluated so pop 27 and 3 and divide 27/3=9 and push it to the stack

Step-8: Now since there is no other character to read from the expression so 9 is the final answer.Pop it.

**Complexity:**

Time Complexity: O(n)

where, n=length of the postfix expression