Given string S representing a postfix expression, the task is to evaluate the expression and find the final value. Operators will only include the basic arithmetic operators like *, /, + and -.

Example 1:

```
Input: S = "231*+9-"
Output: -4
Explanation:
After solving the given expression,
we have -4 as result.
```

Example 2:

```
Input: S = "123+*8-"
Output: -3
Explanation:
After solving the given postfix
expression, we have -3 as result.
```

Your Task:

You do not need to read input or print anything. Complete the function evaluatePostfixExpression() that takes the string S denoting the expression as input parameter and returns the evaluated value.

Expected Time Complexity: O(|S|)

Expected Auixilliary Space: O(|S|)

Constraints:

1 ≤ |S| ≤ 105

0 ≤ |Si|≤ 9 (And given operators)

Solution:

```
class Solution {
    public static boolean isOperator(Character c) {
        return c == '*' || c == '/' || c == '+' || c == '-';
    }

    // Function to evaluate a postfix expression.
    public static int evaluatePostFix(String S) {
        Stack<Integer> st = new Stack<>();
        int n = S.length();

        for (int i = 0; i < n; i++) {
            char c = S.charAt(i);

            if (isOperator(c)) {
                int second = st.pop();
                int first = st.pop();

                switch (c) {
                    case '*':
                        first *= second;
                        break;
                    case '/':
                        first /= second;
                        break;
                    case '+':
                        first += second;
                        break;
                    case '-':
                        first -= second;
                        break;
                }

                st.push(first);
            } else {
                st.push(Character.getNumericValue(c));
            }
        }

        return st.pop();
    }
}
```