14th Jan 2023

First :-

Infix to Postfix : : Medium

Given an infix expression in the form of string **str**. Convert this infix expression to postfix expression.

- **Infix expression:** The expression of the form a **op** b. When an operator is in-between every pair of operands.
- **Postfix expression:** The expression of the form a b **op**. When an operator is followed for every pair of operands.
  Note: The order of precedence is: ^ **greater than** * **equals to** / **greater than** + **equals to** -.

Example 1:

```
Input: str = "a+b*(c^d-e)^(f+g*h)-i"
Output: abcd^e-fgh*+^*+i-
Explanation:
After converting the infix expression
into postfix expression, the resultant
expression will be abcd^e-fgh*+^*+i-
```

Example 2:

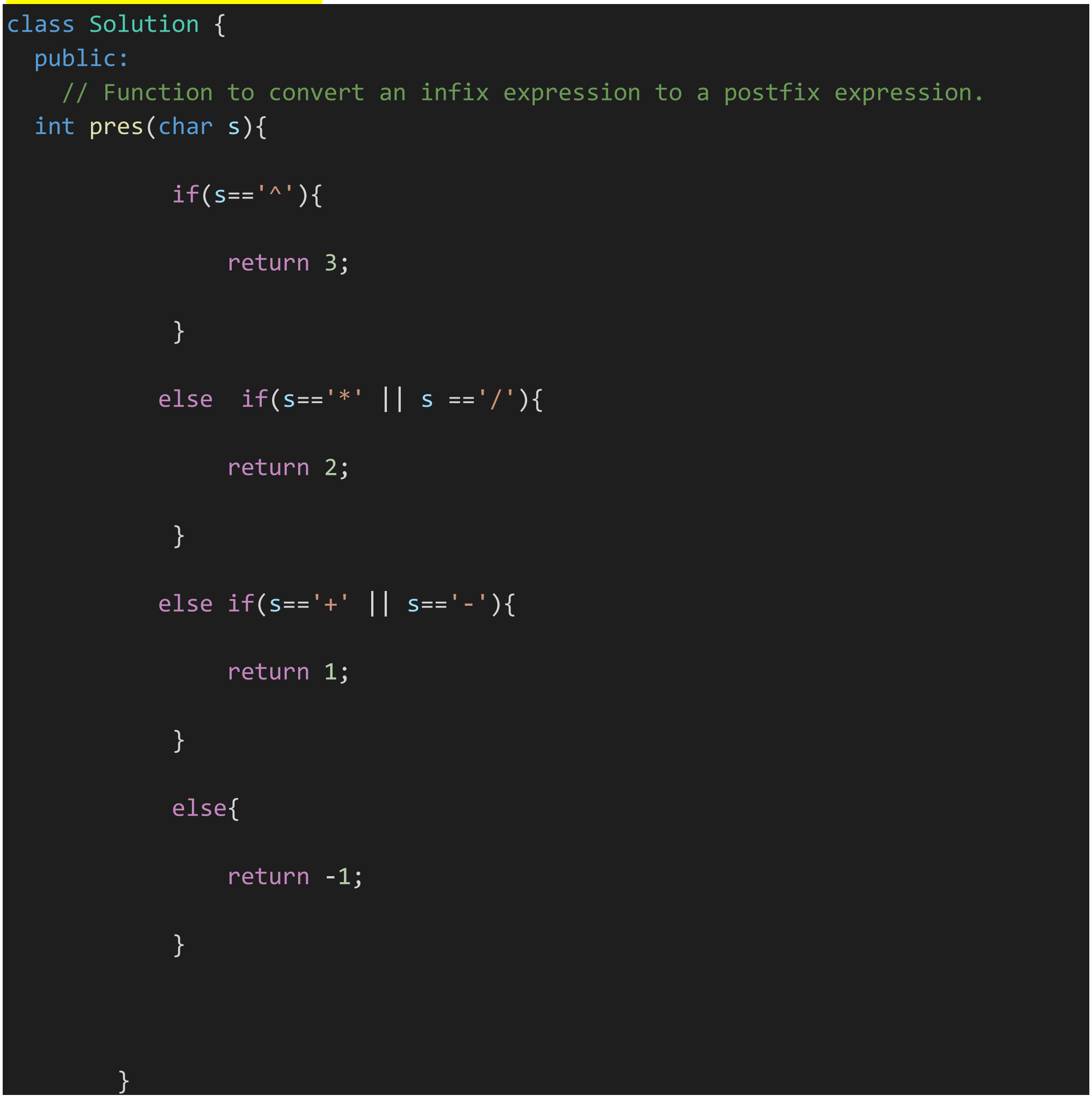
```
Input: str = "A*(B+C)/D"
Output: ABC+*D/
Explanation:
After converting the infix expression
into postfix expression, the resultant
expression will be ABC+*D/
```

Your Task:
This is a **function** problem. You only need to complete the function **infixToPostfix()** that takes a **string**(Infix Expression) as a **parameter** and **returns** a **string(**postfix expression**)**. The **printing** is done **automatically** by the **driver code**.

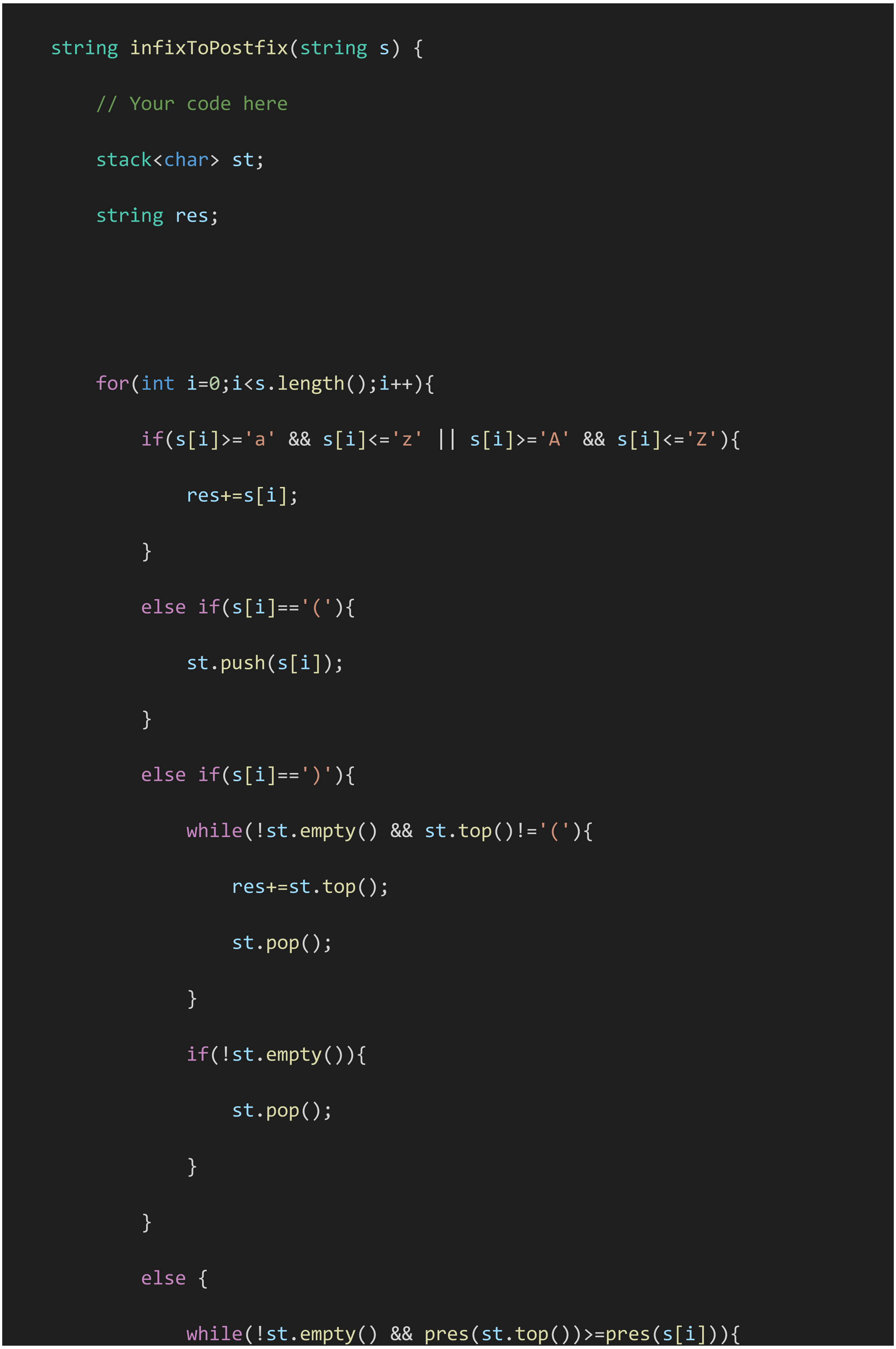
Expected Time Complexity: O(|str|).
Expected Auxiliary Space: O(|str|).

Constraints:
1 ≤ |str| ≤ 10^5

CODE SECTION:-



```
class Solution {
  public:
    // Function to convert an infix expression to a postfix expression.
  int pres(char s){

            if(s=='^'){

                return 3;

            }

           else  if(s=='*' || s =='/'){

                return 2;

            }

           else if(s=='+' || s=='-'){

                return 1;

            }

            else{

                return -1;

            }



        }
```



```
string infixToPostfix(string s) {
    // Your code here
    stack<char> st;
    string res;



    for(int i=0;i<s.length();i++){
        if(s[i]>='a' && s[i]<='z' || s[i]>='A' && s[i]<='Z'){
            res+=s[i];
        }
        else if(s[i]=='('){
            st.push(s[i]);
        }
        else if(s[i]==')'){
            while(!st.empty() && st.top()!='('){
                res+=st.top();
                st.pop();
            }
            if(!st.empty()){
                st.pop();
            }
        }
        else {
            while(!st.empty() && pres(st.top())>=pres(s[i])){
                    res+=st.top();

                    st.pop();

                }

                st.push(s[i]);

            }

        }

        while(!st.empty()){

            res+=st.top();

            st.pop();

        }

        return res;
    }
};
```

-: DONE FOR THE DAY :-