```
#include <iostream>
#include <stdio.h>
#include <string>
#include <ctype.h>
#include <stdlib.h>
using namespace std;
class Node{
public:
    int data;
    Node* prev;
};
class stack{
    Node *top;
public:
    stack(){
        top=NULL;
    }
    void push(int n){
```

```
    Node *temp;
    temp=new Node();
    temp->data=n;
    temp->prev=top;
    top=temp;
}
int pop(){
    int a;
    if(top==NULL){
        cout<<"UNDERFLOW\n";
    }
    else{
        a=top->data;
        top=top->prev;
        return a;
    }
}
int empty(){
        if(top!=NULL)
            return 0;
        else
                return -1;
    }
    void  display(){
        Node* stack1=top;
        while(stack1!=NULL){
            cout<<stack1->data<<" ";
            stack1=stack1->prev;
        }
    }
};
class Test {
    // Method to evaluate value of a
postfix expression
public:
    int evaluatePostfix(string exp) {
        //create a stack
        stack stack1;
        auto st=exp.c_str();
        // Scan all characters one by one
        for (int i = 0; i < exp.length(); i++) {
            char c = st[i];
            // If the scanned character is an
operand (number here),
            // push it to the stack.
            if (isdigit(c))
                stack1.push(c - '0');
                //  If the scanned character is
an operator, pop two
                // elements from stack apply
the operator
            else {
                int val1 = stack1.pop();
                int val2 = stack1.pop();
            switch (c) {
                case '+':
                    stack1.push(val2 + val1);
                    break;
                case '-':
                    stack1.push(val2 - val1);
                    break;
                case '/':
                    stack1.push(val2 / val1);
                    break;
                case '*':
                    stack1.push(val2 * val1);
                    break;
            }
        }
    }
    return stack1.pop();
}
```

```
};
// Driver program to test above
functions
int main(){
   Test t;
   string exp="231*+9-";
   cout<<t.evaluatePostfix(exp);
}
```