

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
struct Node {
```

```
int data;
```

```
Node* next;
```

```
};
```

```
class Stack {
```

```
private:
```

```
Node* top;
```

```
public:
```

```
Stack() : top(nullptr) {}
```

```
void push(int value) {
```

```
Node* newNode = new Node{value, top};
```

```
top = newNode;
```

```
}
```

```
int pop() {
```

```
if (top) {
```

```
int value = top->data;
```

```
Node* temp = top;
```

```
top = top->next;
```

```
delete temp;
```

```
return value;
```

```
}
```

```
return 0;
```

```
}
```

```
bool isEmpty() {
```

```
return top == nullptr;
```

```
}
```

```
};
```

```
int evaluatePostfix(const string& postfix) {
```

```
    Stack stack;
```

```
    for (char ch : postfix) {
```

```
        if (isdigit(ch)) {
```

```
            stack.push(ch - '0');
```

```
        } else {
```

```
            int b = stack.pop();
```

```
            int a = stack.pop();
```

```
            switch (ch) {
```

```
                case '+': stack.push(a + b); break;
```

```
                case '-': stack.push(a - b); break;
```

```
                case '*': stack.push(a * b); break;
```

```
                case '/': stack.push(a / b); break;
```

```
            }
```

```
}
```

```
}
```

```
return stack.pop();
```

```
}
```

```
int main() {
```

```
    string postfix;
```

```
    cout << "Enter postfix expression: ";
```

```
    cin >> postfix;
```

```
    cout << "Result: " << evaluatePostfix(postfix) << endl;
```

```
    return 0;
```

```
}
```