

```
1 #include <stack>
2 #include <queue>
3 #include <iostream>
4
5 using namespace std;
6
7 template<typename T>
8 class Ventana{
9     private:
10         stack<T> pila1;
11         stack<T> pila2;
12     public:
13         void insertar(T nuevo)
14         {
15             pila1.push(nuevo);
16         }
17         void moverIzquierda()
18         {
19             if(!pila1.empty())
20             {
21                 pila2.push(pila1.top());
22                 pila1.pop();
23             }
24         }
25         void moverDerecha()
26         {
27             if(!pila2.empty())
28             {
29                 pila1.push(pila2.top());
30                 pila2.pop();
31             }
32         }
33         void borrar()
34         {
35             if(!pila1.empty())
36             {
37                 pila1.pop();
38             }
39         }
40         friend ostream& operator<<(ostream &f, const Ventana<T>& arg)
41         {
42             stack<T> aux1=arg.pila1;
43             stack<T> aux2=arg.pila2;
44             stack<T> aux3;
45             while(!aux1.empty()){
46                 aux3.push(aux1.top());
47                 aux1.pop();
48             }
49             while(!aux3.empty()){
50                 f << aux3.top() ;
51                 aux3.pop();
52             }
53             while(!aux2.empty()){
54                 f << aux2.top() ;
55                 aux2.pop();
56             }
57             return f;
58         }
59 };
60
```

```
61
62 int main(){
63
64     Ventana<char> prueba;
65
66     prueba.insertar('A');
67     prueba.moverIzquierda();
68     prueba.insertar('B');
69     prueba.insertar('8');
70     prueba.insertar('C');
71     prueba.borrar();
72     prueba.moverDerecha();
73     prueba.insertar('E');
74     prueba.insertar('F');
75     cout << prueba;
76
77
78
79
80 }
```