```cpp
#include <stack>
#include <queue>
#include <iostream>

using namespace std;

template<typename T>
class Ventana{
    private:
        stack<T> pila1;
        stack<T> pila2;
    public:
        void insertar(T nuevo)
        {
            pila1.push(nuevo);
        }
        void moverIzquierda()
        {
            if(!pila1.empty())
            {
                pila2.push(pila1.top());
                pila1.pop();
            }
        }
        void moverDerecha()
        {
            if(!pila2.empty())
            {
                pila1.push(pila2.top());
                pila2.pop();
            }
        }
        void borrar()
        {
            if(!pila1.empty())
            {
                pila1.pop();
            }
        }
        friend ostream& operator<<(ostream &f, const Ventana<T>& arg)
        {
            stack<T> aux1=arg.pila1;
            stack<T> aux2=arg.pila2;
            stack<T> aux3;
            while(!aux1.empty()){
                aux3.push(aux1.top());
                aux1.pop();
            }
            while(!aux3.empty()){
                f << aux3.top() ;
                aux3.pop();
            }
            while(!aux2.empty()){
                f << aux2.top() ;
                aux2.pop();
            }
            return f;
        }
};

```

```cpp

int main(){

    Ventana<char> prueba;

    prueba.insertar('A');
    prueba.moverIzquierda();
    prueba.insertar('B');
    prueba.insertar('8');
    prueba.insertar('C');
    prueba.borrar();
    prueba.moverDerecha();
    prueba.insertar('E');
    prueba.insertar('F');
    cout << prueba;




}
```