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

using namespace std;

template <typename T>

class queue2{

    private:
        stack<T> frente;
        stack<T> atras;

    public:
        //vacio
        bool empty() const{
            return this->frente.empty();
        }
        //tamaño
        int size() const{
            return this->frente.size();
        }
        //front
        const T& front() const {
            return this->atras.top();
        }
        //back
        const T& back() const {
            return this->frente.top();
        }
        //push
        void push(const T & nuevo){
            this->frente.push(nuevo);
            stack<T> aux,aux2;
            aux2=this->frente;
            while(!aux2.empty()){
                aux.push(aux2.top());
                aux2.pop();
            }
            this->atras=aux;
        }
        //pop
        void pop(){
            this->atras.pop();
            stack<T> aux,aux2;
            aux2=this->atras;
            while(!aux2.empty()){
                aux.push(aux2.top());
                aux2.pop();
            }
            this->frente=aux;
        }
};

int main(){
    queue<int> una;

    una.push(5);
    una.push(7);
```

```
61      una.push(3);
62      una.push(5);
63      una.push(8);
64      una.push(3);
65
66      while(!una.empty())
67      {
68          cout<<una.front()<<" ";
69          una.pop();
70      }
71      cout << endl;
72      queue2<int> otro;
73
74      otro.push(5);
75      otro.push(7);
76      otro.push(3);
77      otro.push(5);
78      otro.push(8);
79      otro.push(3);
80      cout << otro.size() << endl;
81
82      while(!otro.empty())
83      {
84          cout<<otro.front()<<" ";
85          otro.pop();
86      }
87  }
```