```
#include <iostream>
#include "GenericList.cc"

using namespace std;

template <class qtype>
class Queue
{
        private:
                List<qtype> qlist;
        public:
                Queue() {}

                Queue(const Queue &q)
                {
                        *this = q;
                }

                void operator=(const Queue &q)
                {
                        qlist = q.qlist;
                }

                void Enqueue(qtype data)
                {
                        qlist.Last();
                        qlist.InsertAfter(data);
                }

                qtype Dequeue()
                {
                        qtype temp = Peek();
                        qlist.Remove();
                        return temp;
                }

                qtype Peek()
                {
                        qlist.First();
                        return qlist.GetValue();
                }

                int Size()
                {
                        return qlist.GetSize();
                }

                bool IsEmpty()
                {
                        return qlist.IsEmpty();
                }

                bool IsFull()
                {
                        return qlist.IsFull();
                }

                Queue operator+(const Queue &q)
                {
                        Queue nu;
                        nu.qlist = qlist + q.qlist;
                        return nu;
                }

                bool operator==(const Queue &q)
                {
                        return (qlist == q.qlist);
                }

                bool operator!=(const Queue &q)
                {
                        return !(*this == q);
                }

                friend ostream& operator<<(ostream& out, const Queue& q)
                {
                        out << q.qlist;
                        return out;
                }
};
```