```
#include "IntQueue.h"

IntQueue::IntQueue() {
    size = 0;
    front = back = NULL;
}

bool IntQueue::isEmpty() const {
    return front == NULL;
}

bool IntQueue::isFull() const {
    return false;
}

bool IntQueue::dequeue(int& frontItem) {
    if ( isEmpty() )
        return false;

    frontItem = front->elem;
    Node * old = front;
    front = old->next;
    delete old;
    size--;
    return true;
}

bool IntQueue::enqueue(const int& x) {
    if ( isEmpty() )
        front = back = new Node(x);
    else {
        back->next = new Node(x);
        back = back->next;
    }
    back->elem = x;
    size++;
    return true;
}

int IntQueue::length() const {
    return size;
}

std::string IntQueue::display() {
    int counter = 0;
    std::stringstream elem;
    std::string elements;
    Node * helper = front;

    while (counter < size)
    {
        elem << helper->elem << ", ";
        helper = helper->next;
        counter++;
    }

    elements = elem.str().substr(0, elem.str().size()-2);
    return elements;
}
```