```
// Stefan Retief
// CS211 DLL, HW7
// DLL Class - Header File
// =============================================================

#ifndef DLL_hpp
#define DLL_hpp
#include <string>
using namespace std;

typedef int el_t;

struct node {                    // stuct that holds the element and a pointer to
    the next element
    el_t elem;
    node* next;
    node* prev;
};

class DLL {
private:
    node* front;                 // address to the front element
    node* rear;                  // address to the last element
    //int count;                   // how many elements do we have right now?
    void queueError(string);     // This displays an error messages passed to it
        and does exit(1);
public:
    int count;
    DLL();                    // constructor DLL();
    ~DLL();                   // deconstructor DLL();

    //HOW TO CALL:  CALL onto an existing DLL ([DLL].addRear(elem))
    //              with the element you want to add to the DLL
    //PURPOSE:      to add a pointer with an element to te list
    void addRear(el_t elem);

    //HOW TO CALL:  CALL onto an existing DLL ([DLL.deleteFront())
    //PURPOSE:      Deletes the rear pointer and returns the element
    el_t deleteFront();

    //HOW TO CALL:  CALL onto an existing DLL ([DLL.deleteRear())
    //PURPOSE:      Deletes the front pointer and returns the element
    el_t deleteRear();

    //HOW TO CALL:  CALL onto an existing queue ([DLLName].isEmpty())
    //PURPOSE:      Checks if the queue has no (0) elements and returns
    //              true or false accordingly
    bool isEmpty();

    //HOW TO CALL:  CALL onto an existing queue ([DLLName].displayADLL())
    //PURPOSE:      Displays aDLL elements in the DLL
    void displayAll();

    void printAllReverseDLL();

    //HOW TO CALL:  CALL onto an existing queue with the element you want
    //              to add as a parameter ([DLLName].addFront(elem))
    //PURPOSE:      To add an element to the front of the DLL
    void addFront(el_t);

    //HOW TO CALL:  CALL onto an existing queue with the element you want
    //              to delete as a parameter ([DLLName].deleteNode(elem))
    //PURPOSE:      To delete the first node containing the element
    void deleteNode(el_t);

    //HOW TO CALL:  CALL onto an existing queue with the element you want
    //              to delete as a parameter ([DLLName].deleteNodes(elem))
    //PURPOSE:      To delete the aDLL the nodes containing the element
    void deleteNodes(el_t);

    //HOW TO CALL:  CALL onto an existing DLL to add en element in Ascending
    //              order (DLLNAME].addInOrderAscend(elem))
    //PURPOSE:      To add element from lowest to greatest
    void addInOrderAscend(el_t e);

    //HOW TO CALL:  CALL onto an existing DLL to add en element in descending
    //              order (DLLNAME].addInOrderAscend(elem))
    //PURPOSE:      To add element from greatest to lowest
    void addInOrderDescend(el_t e);

    //HOW TO CALL:  CALL onto an existing DLL to add an element to see if the
    //              element exists in the list
    //PURPOSE:      To find the element in the list
    bool search(el_t e);

};



#endif /* DLL_hpp */
```