

## CSE 202 - Homework 4

### Templates

Using a template, create a class *Mlist* which will handle different type objects. An *mlist* is a list of objects (numbers, strings, etc) which has functions which allow additions to the front, and back of the list, removals from the front and back of the list, and removals from anywhere in the list. The accessor functions will return the front and end of the list, and will tell if an object is in the list. We will use iterators as well as recursion to implement the class.

Begin by creating the template class with the following interface :

```
template<typename T>
class Mlist
{
public:
    Mlist();//creates the list
    T front();//returns the front of the list
    T end();//returns the end of the list
    bool in(T x);//returns true if x is in the list and false otherwise
    bool empty(); }// returns true if the list is empty
    void addfront(T entry);//add entry to the back of the list
    void addend(T entry);//add entry to the back of the list
    void addorder(T entry);//add entry to an ordered list
    void removefront();//removes the front of the list
    void removeend();//removes the back of the list
    void remove(T n);//searches the list and removes the entry with value n
private:
    vector<T> mlist;
    void remove(typename vector<T>::iterator ix, T n);//uses an iterator and recursion to remove value n
    void addorder(typename vector<T>::iterator ix, T n);//uses an iterator and recursion to add value n in an ordered list
}; // Mlist
```

Create and implement the template class *Mlist*. INCLUDE a main program that creates an *Mlist* and tests each function.