CSE 202 - Laboratory 10

Templates

In this lab, we are going to build most of the template class needed for Homework Assignment 4.

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. We will use iterators as well as recursion to implement the class.

1. Begin by creating the template class with the following interface :

```
template<class 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
```

2. Now write the member functions Mlist, addfront, addend, removefront, removeend, etc.. The function remove is a bit tricky. Use the recursive method discussed in class. You should use an iterator here.
3. Now create a main program to test your template. Use something like :

```
int main()
{
 Mlist<int> test1=Mlist<int>() ;
 test1.add(5);
 test1.add(7);
 test1.add(4);
 test1.remove(7);
 cout << test1.front()<< endl;
 cout << test1.back()<< endl;
 Mlist<string> test2= Mlist<string>() ;
 test2.add("John");
 test2.add("Paul");
 test2.add("Mary");
 test2.add("Kate");
 test2.remove("Paul");
 cout << test2.front()<< endl;
 cout << test2.back()<< endl;
}
```