

- (15 points) I have web pages all over my account. Each has a unique name, and has an .html extensions. I would like to copy all of my web pages to a directory named Web in my home directory. Write a UNIX command line that will accomplish this task. You may assume that I am in the home directory of my account. You should also be aware that there are directories with names that end with "html".

For the rest of this exam you will provide the appropriate C++ code to make the code on the handout work properly. The program reads in events from a user, and stores them in a double linked list. When the user is done entering events, the list saves them into a file. Note that the ListNodes actually store pointers to the data type (Date\* in this program). There is no need for comments.

- (22 points) Provide the implementation code for the save method of the List class that would be in list.h. The method writes to the specified file, and deletes the ListNodes. The header is provided. You may wish to do #4 first.

```
void save(const char *filename) {
```

- (66 points) Provide the complete DateTime.h file including all preprocessor directives appropriate for a large multi-file program. The file should include all of the code necessary to implement the DateTime class. The DateTime class is derived from the Date class, and contains an int hour, and int minute. The DateTime class has two methods, a normal constructor and print(). Note that your print() method must format the line to match the output in the example.

```
1234567890123456789
    June  7  8:03
```

- (57 points) Provide the implementation code for the List insert method that would appear in list.cpp. Beware that you are dealing with pointers to Dates. The list is sorted using the operator<.

```
int main(int argc, char* argv[])
{
    List <Date> list;
    char month[10];
    int day, hour, minute;
    cout << "Event: ";
    cin >> month >> day >> hour >> minute;

    while (day > 0)
    {
        if(hour >= 0)
            list.insert((Date*) new DateTime(month, day, hour, minute));
        else
            list.insert(new Date(month, day));

        cout << "Event: ";
        cin >> month >> day >> hour >> minute;
    } // while day > 0

    list.save(argv[1]); // argv[1] contains the name of the file
    return 0;
}
```

```
[ssdavis@lect1]$ a.out test.txt
Event: April 17 -1 0
Event: February 27 12 0
Event: February 29 23 59
Event: October 31 -1 0
Event: June 7 8 3
Event: June 16 18 10
Event: January -1 0 0
[davis@lect15 mid2b]$
```

```
[ssdavis@lect1]$ cat test.txt
February 27 12:00
February 29 23:59
April 17
June 7 8:03
June 16 18:10
October 31
[davis@lect15 mid2b]$
```

**list.h** contains:

```
template <class T>
class ListNode {
    T *data;
    ListNode<T> *prev;
    ListNode<T> *next;
    ListNode(T *d, ListNode<T> *p, ListNode<T> *n);
    friend class List<T>;
}; // class ListNode

template <class T>
class List
{
    ListNode <T> *head;
public:
    List();
    void insert(T *d); // sorted using Date < operator
    void save(const char *filename);
}; // class List
```

**date.h** contains:

```
class Date
{
    char month[10];
protected:
    int day;
public:
    Date(const char *m, int d);
    const char* getMonth()const;
    bool operator< (const Date &d) const;
    virtual void print(ostream &os)const;
}; // class Date
```