CS 61B, Spring 1996
Midterm #1
Professor M. Clancy

Problem 0 (1 point, 1 minute)

Put your login name on each page. Also make sure you have provided the information requested on the first page.

Problem 1 (4 points, 10 minutes)

A program to print a calender as in programmang assignment 2 includes the following code.

```
   for(int month=JAN; month<=DEC; month++) {
      PrintHeading (month, year);
      PrintDates (month, year);
   }
```

The PrintDates function includes the following code.

```
   int firstDay = DayOfWeek (month, 1, year);
   for (int k=1; k<=firstDay; k++) {
      cout << "   ";
   }
   for (int date=1; date<=NumDaysIn (month, year); date++) {
      cout << setw(3) << date;
      if (IsEndOfWeek (date, firstDay)) {
         cout << endl;
      }
   }
```

Suppose that when the program is run, a calendar is printed in which every month starts on a Sunday as shown below.

```
    January 1996

      S   M   T   W   T   F   S
      1   2   3   4   5   6   7
      8   9  10  11  12  13  14
     15  16  17  18  19  20  21
```

```
      22  23  24  25  26  27  28
      29  30  31

     February 1996

       S   M   T   W   T   F   S
       1   2   3   4   5   6   7
       8   9  10  11  12  13  14
      15  16  17  18  19  20  21
      22  23  24  25  26  27  28
      29

     March 1996

       1   2   3   4   5   6   7
       8   9  10  11  12  13  14
      15  16  17  18  19  20  21
      22  23  24  25  26  27  28
      29  30  31
        ...
```

Part a
Which of the following functions is most likely to contain the bug? Circle the name of the probably buggy function below, and briefly describe a flaw in the function that could produce the incorrect output.

function most likely to contain the bug	*flaw in the function that would cause the incorrect output*
PrintHeading	
PrintDates	
NumDaysIn	
IsEndOfWeek	
DayOfWeek	

Part b
List the emacs or gdb commands you would type after starting gdb to verify your guess from part a.

Problem 2 (4 points, 10 minutes)

Each of the C++ program segements below is part of a .cc file that #includes the intlists.h file on the next page. The last statement given in each program segment is incorrect in some way. For each segment, do the following:

- if the g++ compiler would generate an error message for the last statement in the segment, describe the error message;
- if the g++ compiler would not generate an error message for the segment, write "no g++ msg" and describe what's wrong with the segment in your own words.

file and statement	g++ *error message*
in a main program: `List list2 = new List;`	
in an incorrect version of intlists.cc: `ListNode* ptr = new ListNode;`	

in an incorrect version of intlists.cc:
`void Print (List list5) {`
`   ListNode* ptr = head;`
`      ...`
`}`	

in an incorrect List member function in intlists.cc:
`ListNode* ptr;`
`ptr->next = 0;`	

Abbreviated intlists.h

```
typedef int InfoType;

struct ListNode;         // defined in implementation file

class List {
public:
   List ();              // initializes an empty list
   List (const List&);   // copy constructor
   ~List ();             // destructor
```

```
      // See if the list object is empty.
      bool IsEmpty ();

      // Return the list object's lenght.
      int Length ();

      // Note in the following that positions in the list
      // start with 1 (unlike in Scheme).

      // Insert a new item so that it's at a given position in the
      // list object (positions of subsequent items increase by 1).
      // success is set to false if position was out of range.
      void Delete (int position, bool& success);

      // Return the item at a given position in the list object.
      // success is set to false if position was out of range.
      void Retrieve (int position, InfoType& item, bool& success);

      // Add an item to the end of the list object.
      void AddToEnd (InfoType newLastItem);

   private:
      ListNode* PtrTo (int position);
      int size;
      ListNode* head;       // pointer to first thing in the list
   };
```

First part of intlists.cc

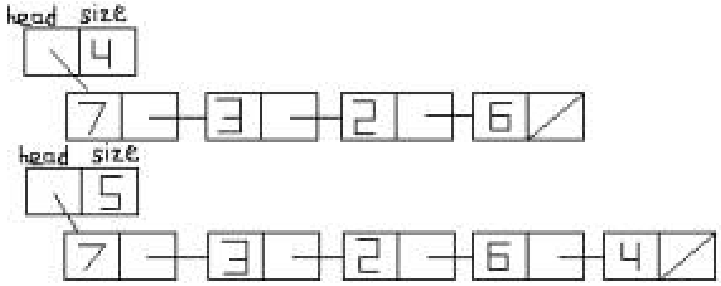
```
   struct ListNode {
      InfoType info;
      ListNode* next;

      ListNode (InfoType item);    // info item and next 0
      ListNode (InfoType item, ListNode* ptr);
   };
```

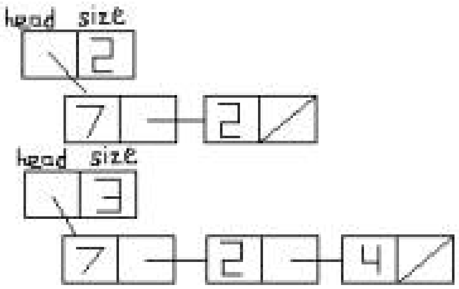
Problem 3 (4 points, 10 minutes)

Fill in the blanks in the List member function RemoveAlternateElements whose framework appears below. RemoveAlternateElements should remove and delete alternate nodes in the List object starting with the second node in the list. If the List object is empty or contains only one element, RemoveAlternateElements should leave it unchanged. Two examples of how RemoveAlternateElements should work with longer lists appear below.

list before call to RemoveAlternateElements



list after call to RemoveAlternateElements



Don't change any of the framework code. Also, don't include any calls to other List member functions.

```
   void List::RemoveAlternateElements() {
      if(head != 0 && head->next != 0) {
         for (ListNode* ptr=head; ptr!=0 && ptr->next!=0; ptr=ptr->next) {



         }
      }
   }
```

Problem 4 (4 points, 10 minutes)

Recode the RemoveAlternateElements so that it is *not* a List member function, but instead calls List member functions declared in the intlists.h file on page 5.

Problem 5 (3 points, 9 minutes)

Consider the following `make` file.

```
vendmain: vending.o vendmain.o
       g++ -g vending.o -o vendmain.o
vending.o: vending.cc vending.h
       g++ -g -Wall -fno-builtins vending.cc -c
vendmain.o: vendmain.cc vending.h
       g++ -g -Wall -fno-builtins vendmain.cc -c
```

Part a

suppose the command "make vendmain" is given in a working directory containing only the files vending.cc, vending.h, and vendmain.cc. What happens? (Hint: vendmain is *not* successfully created.)

Part b

Fix the make file so that the command "make vendmain" does successfully create the executable vendmain.

Posted by HKN (Electrical Engineering and Computer Science Honor Society)
University of California at Berkeley
If you have any questions about these online exams
please contact mailto:examfile@hkn.eecs.berkeley.edu