Midterm Examination

COS 217, Spring 2004

There are 4 problems on this examination, each worth from 10 to 40 points. Each problem should take you about *half* its point value in minutes to complete. You have 50 minutes to complete this examination.

The examination is open book and open notes. You may use any of the printed material associated with this semester's course: required and recommended textbooks, lecture notes, precept handouts, your assignment solutions, etc. However you may not use material that is not directly associated with this semester's course, and you may not use calculators or other electronic devices.

You are not required to comment your code in this exam unless what you have written is difficult to understand.

Please start your solution to each problem on a right-hand side page in your examination booklet.

Partial credit will be given if you can show your work in arriving at solutions. Be brief.

Good luck!

Write out and sign the Honor Code pledge before turning in the test.

"I pledge my honor that I have not violated the Honor Code during this examination."

1. (10 points) Answer each of the following questions (one sentence each):

a) What is the purpose of the built-in C operator `sizeof`?
b) What is the data type returned from `sizeof`?
c) What is the purpose of the library call `malloc`?
d) What is the data type returned from `malloc`?
e) Give an example use of `sizeof.`

2. (20 points) Write a function `void PrintPercent(int x, int y)` which computes and prints the percentage of `x/y` in two decimals. For x = 1 and y = 3, this function should print:

```
The result of 1/3 in percentage is 33.33%.
```

Note that your output needs to print “%.” at the end, and that your code should handle exceptions in the computation gracefully.

3. (30 points) This question has two parts:

a) Write a program to count blanks, tabs and newlines. The program takes input from `stdin` and prints out the results to `stdout`.
b) Write a test plan that includes your test cases and how you plan to automate the test (you are not required to implement the automated test).

4. (40 points) A typical question in Silicon Valley programmer job interviews is to write a program to detect a cycle in a linked list. In this question, we assume that you will design the interface for an ADT and your colleague will implement the operations on the ADT. Your job is to write and add a function to test for cycles into the implementation, design the test plan, and implement the test plan. This question has four parts:

a) Design an interface for a “List” ADT. The interface should declare functions for
   - creating a new List object,
   - freeing a List object,
   - inserting an item (of any type) into a List object, while rejecting the insertion if an item that is "equal to" the given item already exists in the List. Hint: How can a List object store items of any type? Hint: How can a List object compare items of any type? (Note that the List object should compare the items, not simply pointers to the items.)
   - removing an item from a List object, and
   - testing if a List object has cycles. This function should return TRUE if there is a cycle and FALSE otherwise.

   Note that the details should be hidden from the client, thus your interface should not expose detailed data structures.
b) Design an implementation of the List ADT that shows its appropriate data structures. For simplicity, you are required to implement the List object as a singly-linked list. In this implementation, you are only required to implement the function of testing for a cycle (You are not required to implement the other four functions). Use “robust programming” techniques learned from the course.
c) Write a test plan that includes your test cases.
d) Write a sample client test program that calls each function at least once, with normal (i.e. non-erroneous) actual parameters. Note that you are required to implement only a subset of the test cases.