

CS 180: Data Structures, Fall 2011

Homework 2

Due by the start of class on Monday, September 26

1. (a) Fill in the diagram below to represent the underlying memory configuration that is present after the following commands are executed:

```
char a('M');
char *c = new char('N');
char &b(a);
char d(b);
char *f = &d;
```

memory contents	memory address
	262
	263
	264
	265
	266
	267
	268
	269
	270
	271

- (b) Now use the diagram below to update the memory configuration from part (a) after the following 5 commands are executed.

```
b = 'P';
*f = 'R';
char *g = c;
c = f;
a = 'Q';
```

memory contents	memory address
	262
	263
	264
	265
	266
	267
	268
	269
	270
	271

2. Describe a **recursive** algorithm to count the number of nodes in a singularly linked list. Note that I'm not requiring C++ code here, although you are welcome to implement and test your code; I simply want a description of the algorithm, similar to the style in chapter 3, such as code fragment 3.38 on page 140.

3. Extra Credit - C-2.3 from the textbook

Most modern C++ compilers have optimizers that can detect simple cases when it is logically impossible for certain statements in a program to ever be executed. In such cases, the compiler warns the programmer about the useless code. Write a short C++ function that contains code for which it is provably impossible for that code to ever be executed, but your favorite C++ compiler does not detect this fact.