

**2016 Spring COP 3502 Quiz #1**

**Date: 1/26/2016**

**Name:** \_\_\_\_\_

Note: Please answer all coding questions with valid C code. You may assume for any question that the appropriate includes have been made and that `i`, `j` and `k` are declared and ready to use as integer variables (as loop indexes).

**Part I - Dynamic Memory Allocation**

For each of the following questions, you may assume that both `size1` and `size2` are integer variables that have been assigned to values prior to the line of code you are to write.

1) (4 pts) Write a single line of code that dynamically allocates an array of `size1` variables of type `double`. Assign the memory allocated to a pointer called `nums`.

\_\_\_\_\_

2) (2 pts) Write a single line of code that frees the memory allocated in question 1.

\_\_\_\_\_

3) (7 pts) Write a segment of code that dynamically allocates a two dimensional array of characters based on a set of integers read in from standard input. In particular, the first integer, `size1`, will represent the number of strings the array will store. The following `size1` integers will represent the lengths of each of the strings, in order. For example, if the input were 3, 8, 5 and 10, then array ought to be prepared to store three strings of length 8, 5 and 10, respectively. Assign the memory allocated to a pointer called `words`. Please leave room for the null character.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

4) (4 pts) Write a segment of code that frees the memory allocated in question 3.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

5) (8 pts) Consider a situation where we have a variable, `myclass`, of type `struct student**` that is pointing to an array of `size1` pointers of type `struct student*`, each of which is pointing to a single `struct student`. In addition, assume that within each struct, both `name` and `gradePtr` are pointing to memory that is dynamically allocated. Write a segment of code to free all of the memory pointed to by the double pointer `myclass`. (Note: Look on the overhead to see a picture of what the allocated memory looks like.) The struct is defined below.

```
struct student {
    char* name;
    struct grades* gradePtr;
    int ID;
    double balance;
};
```

---

---

---

---

---

6) (5 pts) Using a binary search on a sorted array of 16,000,000 items, how many comparisons are necessary to determine whether or not a value is stored in the array? (Note:  $2^{20} \sim 1,000,000$ .) Briefly explain your reasoning.

---

7) (10 pts) Write a function that takes in a sorted array, `list`, of `length` distinct integers and prints out each positive integer in between 1 and 1,000,000 that does NOT appear on the list. It's guaranteed that all of the values stored in `list` are in between 1 and 1,000,000, inclusive.

```
void printMissing(int* list, int length) {
```

```
}
```