

## COP 3502 Section 17 Quiz #1 - Part B (Linked Lists) Solution

**Date: 5/29/2020**

**Start Time: 2:55 pm EST**

**End Time: 3:20 pm EST**

For all questions on this section (except the last), imagine storing strings as linked lists of nodes that store individual characters. Here is the struct that will be used:

```
typedef struct charnode {
    char c;
    struct charnode* next;
} charnode;
```

The string represented by a linked list of nodes is simply the characters, read in order from the front to the back of the list. The next pointer for the last character in the string will be set to NULL. For each question, you'll be asked to write a function that takes in a pointer to the front of the linked list representing a string and either returning or processing some information about the string.

1) (10 pts) Write a function that takes in a pointer to the front of a linked list storing a string and returns the value of that string. Note if front is NULL the value of the string is 0. We define the value of a string to be the sum of the Ascii values of the characters in the string. Recall that internally, characters are stored as integers which can be added without any casting or special syntax. The function prototype is provided:

```
int stringvalue(charnode* front);
```

### **Solution**

```
int stringvalue(charnode* front) {
    int res = 0;
    while (front != NULL) {
        res += (front->c);
        front = front->next;
    }
    return res;
}
```

**Grading: 1 pt copy prototype, 1 pt set result to 0, 2 pts loop, 3 pts update result, 2 pts go to next node, 1 pt return.**

2) (10 pts) Write a function that takes in a pointer to the front of a linked list storing a string and a single character, mych, counts the number of times mych appears in the string and returns this value.

```
int countChar(charnode* front, char mych);
```

**Solution**

```
int countChar(charnode* front, char mych) {  
  
    int res = 0;  
    while (front != NULL) {  
        if (front->c == mych)  
            res++;  
        front = front->next;  
    }  
    return res;  
}
```

**Grading: 1 pt copy prototype, 1 pt set result to 0, 2 pts loop, 3 pts update result, 2 pts go to next node, 1 pt return.**

3) (5 pts) What common breakfast food can you get at Omlet Bar? **Omlet**

**Grading: Give 5 pts to all who submit the exam, regardless of whether they answered this question.**