

COP 3502 Section 18 Quiz #1 - Part B (Linked Lists) Solution

Date: 5/29/2020

Start Time: 3:55 pm EST

End Time: 4:20 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 prints out that string, followed by the newline character. The function prototype is provided:

```
void printString(charnode* front);
```

Solution

```
void printString(charnode* ptr) {
    while (ptr != NULL) {
        printf("%c", ptr->c);
        ptr = ptr->next;
    }
    printf("\n");
}
```

Grading: 1 pt copy prototype, 2 pts loop, 4 pts print (1 pt printf, 1 pt %c, 2 pts ptr->c), 2 pts go to next node, 1 pt print newline

2) (10 pts) Write a function that takes in a pointer to the front of a linked list storing a string and returns the number of lowercase letters in the string. You may use the function `islower` in the library `ctype.h`. The function returns 1 if its input character is a lower case letter, and 0 otherwise. The function prototype for `islower` is provided below, as well as the function prototype for the function you are to write.

```
/* Even though the prototype has islower taking in an int, it's
meant
to take in a character. If that character is a lower case
letter, 1
is returned. Otherwise, 0 is returned. */
int islower(int arg);

int countLowerCase(charnode* front);
```

Solution

```
int countLowerCase(charnode* front) {

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

Grading: 1 pt copy prototype, 1 pt set res 0, 2 pts loop, 3 pts if check, 1 pt res++, 1 pt go to next node, 1 pt return res

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.