COP 3502 Section 16 Quiz #1 - Part A (Dynamic Memory Allocation)

Date: 5/29/2020

Start Time: 1:30 pm EST
End Time: 1:55 pm EST

Directions: Please type up answers in either a Word Document (.doc, docx) or a Text Document (.txt) and upload your document AND SUBMIT IT to the appropriate assignment in Webcourses COP 3502 Section 16. It is strongly suggested you directly type into a document in your computer and don't recopy the questions due to the time constraints. On the document you submit, put your first and last name in the top left hand corner. On the following line, write "My Quiz 1A Answers", centered. Following that, place your answers, numbered, in order (1, 2, 3).

1) (10 pts) Write three lines of code in C that do the following: (a), declare an integer variable n, (b) read in from standard input an integer and store it in the variable n, (c) dynamically allocate an array of n doubles called array, all set to 0.0. Note, you MUST DO each step, including (c) with a single line of code. (Note: for doubles, if all bits are set to 0, then the value of the number is 0.0 as well.)

2) (10 pts) Included below is a section of code allocating memory dynamically for a data structure. Assume that the struct involved (sodacan) does NOT have any dynamically allocated memory. Properly free the memory that is allocated by the code segment.

```
int n;
scanf("%d", &n);
int* sizes = malloc(n*sizeof(int));
sodacan** cans = malloc(n*sizeof(sodacan*));

for (int i=0; i<n; i++) {
  scanf("%d", &sizes[i]);
  cans[i] = malloc(sizes[i]*sizeof(sodacan));
}
```

3) (5 pts) In the class example cs1list.c, we created a list that dynamically grew and shrank as necessary. What was the rationale for doubling the list size instead of adding 1 when the list filled up?

COP 3502 Section 16 Quiz #1 - Part B (Linked Lists)

Date: 5/29/2020

Start Time: 1:55 pm EST
End Time: 2:20 pm EST

Directions: Please type up answers in either a Word Document (.doc, docx) or a Text Document (.txt) and upload your document AND SUBMIT IT to the appropriate assignment in Webcourses COP 3502 Section 16. It is strongly suggested you directly type into a document in your computer and don't recopy the questions due to the time constraints. On the document you submit, put your first and last name in the top left hand corner. On the following line, write "My Quiz 1B Answers", centered. Following that, place your answers, numbered, in order (1, 2, 3).

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 length of that string. Note if front is NULL that represents a string of length 0. The function prototype is provided:

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

2) (10 pts) Write a void function that takes in a pointer to the front of a linked list storing a string and CHANGES that string to be upper case. (So all lower case letters are converted to upper case letters and all other characters are left the same. Recall that there is a macro in ctype.h called toupper, which takes in a character and returns its uppercase version. (This does exactly what you want - if you pass in something other than a lower case letter, it returns the same exact character it received as input. If you pass it a lower case letter, the corresponding upper case letter is returned.) The function prototype for the function you are supposed to write and toupper is provided. Please use toupper in your code.

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

void convertToUpper(charnode* front);
```

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