CMSC 216 Quiz 4 Worksheet

The fourth quiz for the course will be on Fri, Jul 8. The following list provides more information about the quiz:

- The quiz will be a written quiz (no computer).
- Closed book, closed notes quiz.
- Answers must be neat and legible. **You must use pencil.**
- Check the quiz/exam rules at: **http://www.cs.umd.edu/~nelson/classes/examRules.html**

The following exercises cover the material to be included in this quiz. Solutions to these exercises will not be provided, but you are welcome to discuss your solutions with the TA or instructor during office hours.

Exercises

1. What happens when the following code is executed? Explain briefly.

   ```
   int *p = NULL;
   *p = 20;
   ```

2. Is the following code valid? Explain briefly.

   ```
   int *p;
   p = 0;
   ```

3. Is the following code valid? Explain briefly.

   ```
   int x = 20;
   void *p = &x;
   p = p + 1;
   ```

4. What's the difference between calloc and malloc? Which one is more efficient? Explain briefly.

5. Based on the prototype below, provide two alternative prototypes so we still pass a two-dimensional array of characters. Assume each string has a maximum of 80 characters.

   ```
   void myFunction(char *data[])
   ```

6. Which of the following pointer variables uses the largest number of bytes?

   ```
   int *p;
   char *q;
   ```

7. What situation has occurred after the last assignment of the following code?

   ```
   int *p = malloc(sizeof(int));
   *p = 400;
   p = NULL;
   ```

8. Implement a function that removes the last instance present in a singly-linked list.

9. Implement a recursive function that prints the elements of a singly-linked list in reverse order.

10. Implement a function that takes a string as a parameter and returns a duplicate (allocates memory using malloc).

11. Implement a function that takes as parameter an array of integers and returns a linked list with those integer values.

12. Implement the function append that has the prototype below. The function concatenates all the strings present in the array of strings. You can assume the end of the array is marked by NULL. You need to allocate memory for the resulting string.

    ```
    char* append(char *data[])
    ```

13. The following structures will be used for the questions that follow:

    ```
    #define MAX_LEN 80

    typedef struct {
      char *title;
      int duration;
    } Song;

    typedef struct {
      Song* all_songs;  /* array of songs */
      int number_of_songs;
      char album_name[MAX_LEN + 1];
    } Album;
    ```

    a. Define a method that initializes a Song with using a duration and a string provided. The method will make a copy of the string parameter.

    b. Define a method that will initialize an Album structure with information present in a file. The format of the information in the file is as follows:

       ```
       <albumName>
       <numberOfSongs>
       <titleSong1> <durationSong1>
       <titleSong2> <durationSong2>
       …
       <titleSongN> <durationSongN>
       ```

       You can assume a song's title will not be longer than 80 characters.