1) (10 pts) DSN (Dynamic Memory Management in C)

Suppose we have an array to store all of the holiday presents we have purchased for this year. Now that the holidays are over and all the presents have been given out, we need to delete our list. Our array is a dynamically allocated array of structures that contains the name of each present and the price. The name of the present is a dynamically allocated string to support different lengths of strings. Write a function called delete_present_list that will take in the present array and free all the memory space that the array previously took up. Your function should take 2 parameters: the array called present_list and an integer, num, representing the number of presents in the list and return a null pointer representing the now deleted list. (Note: The array passed to the function may be pointing to NULL, so that case should be handled appropriately.)

```
struct present {
     char *present_name;
     float price;
};

struct present* delete_present_list(struct present* present_list, int
num) {

     int i;

     // Check for null pointers
     // 1 point
     if(present_list == NULL)
        return NULL;

     // Traversing the list to free character arrays
     // 4 points
     for(i = 0; i < num; i++)
        free(present_list[i].present_name);

     // Free the array pointer
     // 3 points
     free(present_list);

     // Return null
     // 2 points
     return NULL;

}
```