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

Suppose we are planning a party and we would like to create an array to store our list of supplies. Currently our list is stored in a text file with the name of each item to be purchased on a line by itself. Write a function called make_grocery_list that reads these items from a file and stores them in a two-dimensional character array. Your function should take 2 parameters: a pointer to the file and an integer indicating the number of grocery items in the file. It should return a pointer to the array of items. Be sure to allocate memory for the array dynamically and only allocate as much space as is needed. You may assume that all of the strings stored in the file representing grocery items are alphabetic strings of no more than 127 characters (so the buffer declared is adequate to initially read in the string).

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

char ** make_grocery_list (FILE *ifp, int numItems) {

     char buffer[128];
     char **list = NULL;
     int i;


     // Grading - 2 pts
     list = malloc(sizeof(char *) * numItems);

     // Grading: 6 pts - 1 pt loop, 1 pt read, 4 pts malloc,
     //                  1 pt strcpy
     for(i = 0; i < numItems; i++) {
               fscanf(ifp, "%s", buffer);
               list[i] = malloc(sizeof(char) * (strlen(buffer) + 1));
               strcpy(list[i], buffer);
     }

     // Grading - 1 pt
     return list;


     // Grading notes - 1 pt off for = vs strcpy, 1 pt off for
     // forgetting the +1 in the inner malloc.





}
```