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

A catalogue of ***apps*** and their price is stored in a text file. Each line of the file contains the name of an app (1-19 letters) followed by its price with a space in between. Write a function called ***makeAppArray*** that reads the *app information* from the file and stores it in an array of app pointers. Your function should take 2 parameters: a pointer to the file containing the app information and an integer indicating the number of ***apps*** in the file. It should return a pointer to the array of ***apps***. An ***app*** is stored in a struct as follows:

```
typedef struct{
  char name[20];
  float price;
} app;
```

Make sure to allocate memory dynamically. The function signature is:

```
app** makeAppArray(FILE* fp, int numApps) {

    app** appArray = (app**)malloc(numApps * sizeof(app*)); //3 pts
    int i;
    for(i=0; i < numApps; i++){
        appArray[i] = (app*)malloc(sizeof(app));    // 2 pts
        fscanf(fp, "%s", appArray[i]->name);  // 2 pts
        fscanf(fp, "%f", &(appArray[i]->price)); // 2 pts
     }
     return appArray; // 1 pt
}
```

Grading notes: the casts aren't necessary, no points off for forgetting to declare i, no points off if only one percent code is wrong, 1 pt off if both percent codes are wrong, take only 1 pt off total if the syntax for reading from a file is incorrect, take off only 1 pt total if they use a dot instead of an array, they can order these statements differently - they can allocate ALL of the space before reading anything in.