

COP 3223 Recitation: Structs

Problem: Common DVDs

By the end of the semester, you and your roommate have become great friends and have mutually decided to share some items, such as your DVDs. What you'd like to do is see which DVD's you have duplicate copies of, and then sell the extra copies on eBay, splitting the proceeds equally. To automate the process of determining how much money you expect to make, you decide to write a computer program. You've already decided on the struct to store information about a single DVD:

```
typedef struct {
    char title[100];
    int runTime;
    int idtag;
    double salePrice;
} DVD;
```

We consider two DVDs to be identical if the titles are identical, as determined by strcmp. (Occasionally the same movie has different versions, so titles match but the other components may be different, but for all practical purposes the actual content of the movies is roughly the same. This is similar to the difference between the 6th and 7th edition of any college textbook, for example. There are differences, but they are largely cosmetic and only created in a shameful attempt to make more money.) Note: to simplify the problem, both you and your roommate are smart enough not to buy multiple copies of a movie with the same title.

Write a function that takes in two arrays of DVDs (representing your roommates' DVDs and your DVDs, respectively), the lengths of both arrays, and returns the sum of the sale prices of all the DVDs in both collections with the identical titles. If both of you share a DVD with the same title but different sale prices (the 7th edition always costs more than the 6th), assume that you'll sell the more expensive copy and keep the cheaper one, in an effort to maximize your return.

To test your function, use the testing framework provided. The function prototype is provided below:

```
// Pre-condition: list1 is an array of length len1, list2 is an array
//                of length len2. No title appears in either list more
//                than once.
// Post-condition: Returns the sum of the sale prices of the common
//                DVDs in both lists. Two DVDs are considered the
//                same if their titles are identical.
double getProfit(DVD* list1, int len1, DVD* list2, int len2);
```

You may write any extra support functions you deem useful.