

CS 1101 - A-term 16

Homework 2 - Itemizations and Simple List Processing

Due: Tuesday, September 13 at 5pm

Read the [expectations on homework](#).

Assignment Goals

- To make sure you can define itemizations and write programs over itemizations
 - To make sure you can define simple lists and write programs over lists
-

The Assignment

Writing Data Definitions for Itemizations (you'll be doing this part in Lab 2)

For weeks, firefighters have been battling wildfires in California. Floods have plagued areas in Louisiana, Missouri, and Texas. In this assignment, you'll create data definitions for various kinds of weather-related phenomena, and write programs over those data definitions.

A storm can be any one of the following (*use the names in bold letters to name your structs*):

- **hurricane** Information required for a hurricane consists of the name of the hurricane, the category (a number between 1 and 5, inclusive), the maximum sustained winds in miles per hour, the velocity of the storm in miles per hour, and the storm's heading (for example, NNW).
- **thunderstorm** The definition of a thunderstorm consists of the number of inches of rainfall, the maximum wind gust in miles per hour, the velocity of the storm in miles per hour, and its heading.
- **fire** A fire is represented by the number of square miles it covers, the number of days it has been raging, and the number of people displaced by the fire.

1. Develop a data definition for each type of storm described above, and a data definition for an itemization for storms. When creating a struct, the order of the fields in the struct should match the order given in the descriptions above. For example, the fields of the struct for a hurricane should be the name, category, maximum wind, velocity, and heading, in that order. *Failure to define the fields in the given order will cause our auto-tester to fail, and you will lose points.*
2. Provide at least one example for each kind of storm.

Ask a lab assistant to check over your work at this point.

Writing Programs for Itemizations

3. Provide the template for *each* data definition you made in Problem 1 (including the itemization).

Ask a lab assistant to check over your templates.

If you're working with a homework partner... You and your partner should review the data definitions and templates you each came up with in Lab 2. Make changes, if necessary, and complete any of the

exercises you didn't get to during the lab. Create your Homework 2 file according to the [naming conventions for homework files](#), and copy/paste your data definitions and templates into your homework file.

If you're working by yourself... copy your Lab 2 work into your Homework 2 file (name your file `yourLastName-hw2`), and continue working on Homework 2.

Everyone... Make sure your function names are exactly the same as the names given in the problems. *Otherwise, our auto-tester will fail, and you will lose points.*

4. Write a function `high-impact?` that consumes a storm and produces a boolean. The function returns true if the storm is a category 4 or 5 hurricane, a thunderstorm with more than 3 inches of rainfall and winds exceeding 60mph, or a fire covering at least 50 square miles.
5. Write a function `change-heading` that consumes a storm and a heading and produces a storm. The storm is returned unchanged if the given storm is a fire. Otherwise, the storm that's produced is a storm the same as the original, except that the heading has been changed to the given heading.

Writing Programs for Simple Lists

In the following problems, use this data definition for `ListOfString`:

```
;; a ListOfString is one of  
;;   empty  
;;   (cons String ListOfString)  
;; interp: ListOfString represents a list of strings
```

6. Develop a function `character-count` that consumes a `ListOfString` and counts the total number of characters in all strings in the list.
7. Develop a function `numeric-strings` that consumes a `ListOfString` and produces a `ListOfString`. The list that's produced contains only those strings from the original list that consist entirely of numeric characters. (Hint: check the DrRacket help desk for various built-in string functions that will help you solve this problem).
8. Develop a function `count-x` that consumes a `ListOfString` and counts the total number of X's (upper and lower case) that occur in all strings in the list. Hint: helper functions are your friend. Check the help desk again for built-in-functions that might be useful.

What to Turn In

Here is the [grading sheet](#) that the graders will be using for Homework 2.

Submit your `.rkt` file to [InstructAssist](#). The name of the project is `Homework 2`. Make sure both partners' names and login names appear at the top of the file in a comment.
