

Session 1 Full Homework Problem Set (Homework 3 in the CS1 course)

Problems used in the session in **red**.

Problem 8 – solution interview, Problem 4 – compare alternative solutions

If you watch TV, you're probably pretty sick of watching television ads, especially now that the presidential campaigns are in full swing. The typical 30-minute TV program consists of 22 minutes of programming and 8 minutes of commercials. In this assignment, you'll create data to represent information about television ads, and write programs that process lists of ads.

1. An Ad consists of whether or not the type of the ad is political (as opposed to a product ad), the name of the product or politician the ad is for, the duration of the ad (in seconds), the cost to produce the ad (in thousands of dollars), whether or not the ad is to be aired nationally (as opposed to locally), the time of day that the ad is to be aired (either daytime, primetime, or off-hour), and the number of times the ad is to be aired.

Write data definitions and provide examples of data for Ad and ListOfAd. The name of your struct should be **ad**. Make sure you define the fields in your struct in the order given above. You should use the data type Boolean to represent the fields for political/product and national/local.

2. Write the templates for Ad and ListOfAd.
3. Write a function `count-political-ads` that consumes a list of ads and produces the number of ads in the list that are classified as political ads.
4. Write a function `any-ads-for?` that consumes a list of ads and a String representing a product name or politician's name, and produces a Boolean. The function returns true if the list contains any ads for the given product or politician.
5. Write a function `primetime-ads` that consumes a list of ads and produces a list of all the ads airing in primetime.
6. Write a function `politicians-sponsoring-ads` that consumes a list of ads and produces a list of strings. The list that is produced contains the names of the politicians who have political ads (it's OK if the resulting list contains duplicate names).
7. Write a function `air-cost` that consumes an Ad and produces a Number. The number produced is the cost of airing the ad, which is determined as follows: airtime is sold in 30-second-spot blocks (so if the duration of the ad is 15 seconds, it would cost half the stated amount, etc.). The cost of a 30-second primetime ad for a national market is \$100,000. The cost of a 30-second primetime ad for a local market is \$5000. A discount of 20% is applied to the cost if the ad is aired in the daytime, and a discount of 50% is applied if the ad is aired during off-hours; the discounts apply to both national and local ads. Finally, the air-cost is multiplied by the number of times the ad is to be aired. (You may assume that all airings of an ad occur in the same market and at the same time of day.)
8. Write a function `campaign-air-cost` that consumes a list of ads and the name of a politician, and produces the total air-cost of all political ads for that politician.
9. Write a function `total-ad-cost` that consumes an Ad and produces the total cost of the ad. The total cost is the sum of the cost of producing the ad and the cost of airing the ad.
10. Write a function `expensive-ads` that consumes a list of ads and a Number. The function produces a list of those ads for which the total ad cost exceeds the given number.