Introduction to Computer Science 2018 Aquinas Hobor

Problem Set #11B (total marks: 4; due: 2:29 pm on Monday, April 9th):

This problem set assumes that you have worked through the book chapter 15 (including **all** of the problems at the end of each chapter). Please take a moment to recall the collaboration policy for this course: all "for-credit" problems must be done entirely on your own (no Internet, no classmates, no friends). **You may ask the tutors or me for help**, consult your class notes, and look at the textbook.

1. (2 marks) You have been recruited to help plan a robbery of the Museum of Valuable Liquids. This legendary Museum contains samples of a wide variety of liquids, from water to mercury to cobra venom. You have been given a list of every item in the Museum; each item on the list names the item along with its total weight and value. For example, one item might be:

   ```
   ("water", 202.1, 25.2)
   ```

   indicating that the water sample in the Museum weighs 202.1kg and is worth SGD 25.2. Technicalities: no two samples in the Museum have the same name; all weights are positive; all values are positive.

   Because of the daring nature of the heist, it is not possible to carry away everything: you have a given maximum capacity in kg. Of course, it's a bad idea to mix cobra venom and milk (among other things), so you'll need some flasks to carry the various liquids out of the Museum. Fortunately, you've acquired some flasks that are so light that you can ignore their weight.

   You do not need to take an entire sample: for example, rather than taking the entire "`water`" sample above, you could instead take half (for a weight of 101.05 kg and value of 12.6 SGD). Of course, you can't take more than the Museum actually has ;-).

   Your task: determine which samples (and how much of each) you should take to maximize the value your robber gang extracts from the Museum. Your answer should be a function

   ```
   rob : (string * float * float) list -> float ->
         (string * float * float) list
   ```

   The input to your function is the list of samples and a maximum capacity (in kg); the output is the list of samples you intend to take, along with the amount (in kg) you wish to take from each and the value of that (portion of which you take of the) sample (in SGD).

   In the file "`Intro CS 11B Assignment.ml`" I've written a program `input_catalog` that reads from the file "`Museum.cat`", which lists various liquids, and outputs a list in the appropriate format for your further processing. You can use this for testing; the function `total_value`, in the same file, will take the output of your plan and determine the total value stolen. For the "`Museum.cat`" file, the following should help you test your `rob` function:

```
let items = input_catalog ();;
let plan = rob items 18.68;; (* We can carry 18.68 kg *)
# total_value plan;;
- : float = 3431.21122 (* The SGD value of what we steal *)
```

Important note: To generate the file `Museum.cat`, please run the `Museum_stock.ml` file. You don't need to worry how that file works, but this way I don't have to create different versions of the file for Mac, Windows, etc. You should only need to run it once.

2. (0 marks) Although I say it above, just in case you missed it, please read Chapter 15 and do all of the problems at the end. Pay particular attention to the "`Random`" library.

3. (2 marks) You may have heard of the famous "Monty Hall" problem: https://en.wikipedia.org/wiki/Monty_Hall_problem. Briefly, there are three doors, behind two of which are goats and one door leading to a car. You are invited to select a door at random. The host then opens one of the doors that you did not select, revealing a goat. You must now choose: should you stay with your original choice, or switch to the remaining unopened door?

   Rather surprisingly to many people, you should switch, and doing so improves your odds dramatically, from 1/3 to 2/3. Many mathematicians, including the great Paul Erdös (1913-1996) had a very difficult time believing this. In fact, Erdös was only convinced after seeing a computer simulation of the problem. This is what you will now do.

   Your task: write `monty : int -> int -> bool -> float`, which takes three parameters, the first of which is the number of doors (in the standard problem, 3), the second of which is the number of trials (we will do at least 1,000,000), the third of which is the strategy (true means switch; false means stay). The function then computes the winning percentage.

   Note: when the host has two or more doors hiding goats, he selects one at random.

Upload the solution in a file named `your_first_and_last_name_11b.ml` to Canvas. Put your name and the assignment number in a `(* comment *)` at the top.