C4M Homework: Homework: Week 2 Level 2

Please submit your solutions to these problems on MarkUs at `https://markus.cdf.toronto.edu/c4m2-2016` using your UTORID to authenticate.

1. The following problems have two purposes: to practice loops/type conversions etc., and also to practice the memory model. Make sure that you understand the choices that you make about whether to return a list from the function, or to pass is as a parameter.

   (a) Write a function `get_int10_from_input()` which can be used to get an integer from input and compute its value times 10. The function should repeatedly ask for input from the user, and, if the input it got from the user is numerical, multiply it by 10 and return it. Here is what a sample interaction with the user could look like:

   ```
   In [21]: get_float_from_input()
   Attempt n. 1: please enter an int: hi
   'hi' is not a numerical value. Trying again...
   Attempt n. 2: please enter an int: how are you feeling today?
   'how are you feeling today?' is not a numerical value. Trying again...
   Attempt n. 3: please enter an int: eleventy billion
   'eleventy billion' is not a numerical value. Trying again...
   Attempt n. 4: please enter an int: 123
   Returning the number 1230
   Out[21]: 1230.0
   ```

   Assume that you only need to try to get a valid integer at most 10 times.

   (b) Now write a function `get_5_ints()` that obtains a list containing 5 integers from input and returns that list

   (c) Now write a function `add_5_ints(int_list)` that takes in an already-existing list, and adds 5 integers that it obtains from input to it. Make sure that you understand how the information obtained from input can be usable outside the function.

2. Write a function that compares the number of search results for two sets of search terms on `yahoo.ca`, and returns the search term that gets more search results. For example, if the function tries to compare `["flu", "fever"]` and `["flu", "hives"]`, it'll get approx. 5 million and 300,000 results respectively, and so should return `["flu", "fever"]`.

3. The previous function doesn't really produce a fair comparison: "hives" appears less frequently than "fever." Write a function that takes in the name of a disease (e.g., `"flu"`), and two lists of symptoms. For each symptom (a single term), we can compute the following:

   $$itf(term) = -log(1/nresults(term)).$$

   $itf(term)$ will be larger if term appears more frequently on the internet.

   Here, $nresults(term)$ is the number of search results for a search term. For a disease d and a list of symptoms $\{term0, term1, term2, term3, ...\}$ we can now compute an adjusted search results score using

   $$nresults(d)/\sum_i itf(term_i)$$

   Write a function to compute this score for two sets of symptoms.