CSCI3110-001 JavaScript Exercises

Note: You will be tested on exercises. It is in your best interest to complete them outside of class.

Answer the following using JavaScript.

1. To maintain one's body weight, an adult human needs to consume enough calories daily to (1) meet the basal metabolic rate (energy required to breathe, maintain body temperature, etc.), (2) account for physical activity such as exercise, and (3) account for the energy required to digest the food that is being eaten. For an adult that weights ***P*** pounds we can estimate the caloric requirements using the following formulas:
   a) Basal metabolic rate: calories required = $70 * (P / 2.2)^{0.756}$
   b) Physical activity: calories required = 0.0385 * *Intensity* * *P* * *Minutes*
      Here, Minutes is the number of minutes spent during the physical activity, and Intensity is a number that estimates the intensity of the activity. Here are some sample numbers for the range of values:

Activity	Intensity
Running 10 mph:	17
Running 6 mph:	10
Basketball:	8
Walking 1 mph:	1

   c) Energy to digest food: calories required = *TotalCaloriesConsumed* * 0.1
      In other words, 10% of the calories we consume go towards digestion.

   i. Write a function that computes the calories required for the basal metabolic rate, taking the person's weight as parameter.
   ii. Write another function that computes the calories required for physical activity, taking as input parameters the intensity, weight, and minutes spent exercising.
   iii. Write another function that accepts as parameters a person's weight, an estimate for the intensity of physical activity, the number of minutes spent performing the physical activity, and the number of calories in one serving of your favorite food. Use the previous functions to calculate and then return how many servings of that food should be eaten per day to maintain the person's current weight at the specified activity level. The computation should include the energy that is required to digest food. You can find estimates of the caloric content of many foods on the web. For example, a double cheeseburger has approximately 1000 calories.
2. Your time machine is capable of going forward in time up to 24 hours. The machine is configured to jump ahead in minutes. To enter the proper number of minutes into your machine, you would navigate to a page that can take a start time (in hours, minutes, and a boolean indicating AM or PM) and a future time (in hours, minutes, and a boolean indicating AM or PM) and calculate the difference in minutes between the start and future time.
   A time is specified in your program with three variables:
   ```
   var hours, minutes, isAM;
   ```
   For example, to represent 11:50PM, you would store:
   ```
   hours = 11; minutes = 50; isAM = false;
   ```
   This means that you need six variables to store a start and future time.

Write a function named `computeDifference` that takes the six variables as parameters that represent the start time and future time. Your function should return the time difference in minutes. For example, given a start time of 11:59AM and a future time of 12:01PM, your page should compute 2 minutes as the time difference. Given a start time of 11:59AM and a future time of 11:58AM, your program should compute 1439 minutes as the time difference (23 hours and 59 minutes).

3. In the land of Puzzlevania, Aaron, Luli, and Baba had an argument over which one of them was the greatest puzzler of all time. To end the argument once and for all, they agree on a duel to the death. Aaron is a poor shooter and only hits his target with a probability of 1/3. Luli is a bit better and hits her target with a probability of 1/2. Baba is an expert marksman and never misses. A hit means a 'kill' and the person hit drops out of the duel.

   To compensate for the inequities in their marksmanship skills, it is decided that the contestants would fire in turns starting with Aaron, followed by Luli, and then by Baba. The cycle would repeat until there was one person standing. And that one person would be remembered as the greatest puzzler of all time.

   a) Write a function to simulate a single shot. It should use the following declaration:
   ```
   function shoot(accuracy)
   ```
   This would simulate someone shooting with the given accuracy by generating a random number between 0 and 1. If the random number is less than accuracy, then the target is hit and the function returns true.
   See https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random

   For example, if Luli was shooting at Baba, this could be invoked as:
   ```
   var isBabaDead = shoot(0.5);
   ```
   b) An obvious strategy is for each person to shoot at the most accurate shooter still alive on the grounds that this shooter is the deadliest and has the best chance of hitting back. Write a second function named `startDuel` that uses the `shoot` function to simulate an entire duel using this strategy. It should loop until only one contestant is left, invoking the `shoot` function with the probability of hitting the target according to who is shooting. The function should return who won the duel.
   c) Invoke the `startDuel` function 1000 times in a loop, keeping track of how many times each contestant wins. Output the probability that each contestant will win when everyone uses the strategy of shooting the most accurate shooter alive.
   d) A counterintuitive strategy is for Aaron to intentionally miss on his first shot. Therefore, everyone uses the strategy of shooting at the most accurate shooter left alive. This strategy means that Aaron is guaranteed to live past the first round, since Luli and Baba will fire at each other. Modify the program to accommodate this new strategy and output the probability of winning for each contestant.

4. (***Sieve of Eratosthenes***) A prime number is any integer greater than 1 that is evenly divisible only by itself and 1. The Sieve of Eratosthenes is a method of finding prime numbers. It operates as follows:
   i. Create a boolean array with all elements initialized to true. Array elements with prime indices will remain true. All other array elements will eventually be set to false.
   ii. Starting with array index 2, determine whether a given element is true. If so, loop through the remainder of the array and set to false every element whose index is a multiple of the index for the element with value true. Then continue the process with the next element with value true. For array index 2, all elements beyond 2 in the array that have indices which are multiples of 2 (indices 4, 6, 8, 10, etc.) will be set to false; for array index 3, all elements beyond element 3 in the array that have indices which are multiples of 3 (indices 6, 9, 12, 15, etc.) will be set to false; and so on.

   When this process completes, the array elements that are still true indicate that the index is a prime number. These indices can be displayed. Read the number of elements from a form and then output the prime numbers.

End of Exercise