

Write a webpage that contains several multi-threading functions that do the following

1. Write a random function that for every second, it generates a pair of numbers (second, value). Second is between 1 and 100, value is any number between 1000 and 10000.

2. Write another function that takes the above generated pair and put this pair in a list L. The |second| variable represents how long this pair will stay in the list, i.e., after |second|, the pair will be auto-discarded if not currently in use. Maximum length of the list is 10000 numbers and if the list is full, the old pair will be discarded for a new coming pair to be put in the list.

3. Having a webpage that displays 20 latest pairs in this list L. Auto-update these 20 pairs every 5 second.

4. Write a multiple thread function that each time this function is called, 10 oldest numbers (in terms of arrival time in the list L) in the list are summed. You need to create a button "Display Sum" on the web page so that when the button is clicked, this function will be called and the sum of the oldest 10 numbers will be displayed on the webpage. Subsequent click on this button will display the sum of the next 10 oldest numbers.

5. Store these sum in a list or another data structure and create a button "Display Median" on this web page. The sum list is added a number every time "Display Sum" button is clicked on step 4. Every time this button "Display Median" is clicked, the web page should quickly display the current median of this sum list. For example, if the current sum list is 1000, 2000, 1200, 1100, the median should be 1150 (which is $(1100+1200)/2$). If the current sum list is 1000, 2000, 1200, the median should be 1200. You can use any fast data structure as you wish for this list.

You can use any fast data structure (list, queue, trees, hash table) as you wish for this exercise. You can use any HTML, javascript, jquery, nodejs, angularjs or any language you are comfortable with for the web page. For the function you can also pick any of your preferred language.