Assignment 3

This assignment is worth 4 marks. Each question is worth 1 mark. Deadline 25th April EOD.

General guidelines

You may assume that the input provided is a valid input. For example, if the procedure expects two positive integers as input, you can assume that the input is always given in the correct form.

These questions do not require more than 5-10 lines per question.

Please do not request additional time for submission as I will be ignoring such email requests.

1. Create a circular buffer of fixed size n, each of the n entries is filled with null. Operations are

   - enqueue x at end of buffer (enq buff-name x).
   - dequeue from the beginning of buffer (deq buff-name).

   Assume buffer can overflow. In that case, the buffer will replace the element added at the very end.

   let apple be a buffer currently having elements $(6, 7, 11, null, null)$

   (enq apple 2) should modify apple to $(6, 7, 11, 2, null)$.

   (deq apple) should modify apple to $(null, 7, 11, 2, null)$.

   If two more enq operations (say 15 and 16) are done, then the circular buffer would be completely filled but the starting element of the circular buffer would be the 2nd element (i. e. 7).

   apple would look like this (16,7,11,2,15).

   If another enq operation (say enq apple 14) happens, causing the cb to overflow, it should replace 7 with 14 and move the start of the buffer to the 3rd element.

   Use set! to implement this operation, so that $O(1)$ number of mutations (modifications) are done on the buffer. Time taken is not a parameter that we intend to optimize.

   apple would now look like this (16,14,11,2,15).

2. Implement an in-place insertion-sort using set!. The procedure name is isort and takes as input a list and returns the sorted output produced using insertion sort algorithm.

3. Create a stream named harmonic. The i^{th} element of the stream should contain the sum of i terms of the harmonic series.

4. Create a stream named prime-mod, The i^{th} element of the stream should contain the sum of first i prime numbers modulo 10007.