FIFO: The page is placed from 1 to memory size one by one, so the earliest page is placed at the beginning of array. Therefore, when the array is full and new page is coming in, the program will delete the page at array[0], and then move the rest of the array left by one. Then the new page will be placed at the end of the array. In this way, the pages are placed in time order.

LRU: In least recently used page replacement algorithm, I repeatedly place the incoming page at the end of the array. When an incoming page is not in the array and the array is not full, it is placed in array by time order. Then when the incoming page is in the array and the array is full, it will be pulled and placed at the end of the array, and the pages after it will be shifted left by one. If the incoming page is not in the array and the array is full, the page at array[0] will be deleted, then the rest of pages will be shifted left by one. The incoming page will be placed at the end of array.

I didn’t get second chance worked☹

workloadGen: This program takes in pureScan.txt, which contains number from 1 to 500, and generates randomScan.txt, which contains