**EECS2040 Data Structure Hw #2 (Chapter 3 Stack/Queue)**

**due date 4/17/2021**

***Format***: Use a text editor to type your answers to the homework problem. You need to submit your HW in an HTML file or a DOCX, pdf file named as **Hw2-SNo.docx, Hw2-SNo.pdf** or **Hw2-SNo.html**, where SNo is your student number. Send the **Hw2- SNo.doc or Hw2-SNo.html** file in eLearn. Inside the file, you need to put the **header and your student number, name (e.g., EE2410 Data Structure Hw #2 (Chapter 3 of textbook) due date 4/17/2021 by SNo, name)** first, and then the **problem** itself followed by your **answer** to that problem, one by one. The grading will be based on the correctness of your answers to the problems, and the **format**. Fail to comply with the aforementioned format (file name, header, problem, answer, problem, answer,…), will certainly degrade your score. If you have any questions, please feel free to ask me.

**Part 1 (50%)**

1. (30%) A linear list is being maintained circularly in an array with front and rear set up as for circular queues.

(a) Obtain a formula in terms of the array capacity, front, and rear, for the number of elements in the list.

Ans: (rear – front +capacity)%capacity

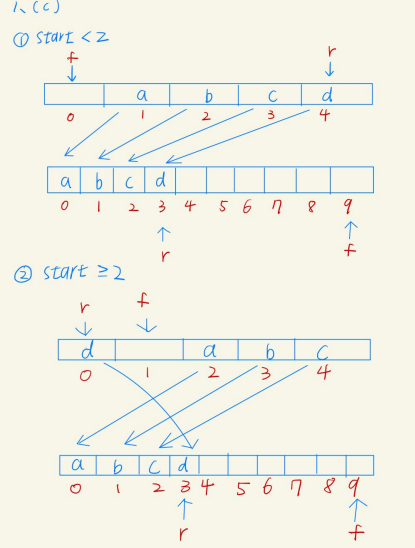
(b) Assume the kth element in the list is to be deleted, the elements after it should be moved up one position. Give a formula describing the positions of those elements to be moved up one position, i.e., from ??? to ???.

Ans: from (front+k+1)%capacity to rear

(c) Assume that we want to insert an element y immediately after the kth element and array doubling is needed (as **Program 3.11** shows or pptx page 83 **void Queue<T>::**Push(**const T&** x) shows), please explain the code using a graphical illustration and explanation.

Ans:

※ 將 circular queue 畫成直的方便觀察



2. (20%) Using the operator priorities of Figure 3.15 (or pptx page 130 **Parentheses Handling)** together with those for ‘(‘ and ‘#’ to answer the following: (a) In function Postfix (Program 3.19, pptx **Infix to Postfix Algorithm**), what is the maximum number of elements that can be on the stack at any time if the

input expression has n operators and delimiters?

Ans:

(��9)\*8 + (n%9) ,if n%9< 7

(��9)\*8 + 7 ,if n%9≥ 7 <= max

(b) What is the answer to (a) if the input expression e has n operators and the depth of nesting of parentheses is at most 6?

Ans: n+6 ,n<49

55 ,n=49 <= max

3. (50%) Write the postfix form and prefix form of the following infix expressions: (a) –A + B – C + D

(b) A \* -B + C

(c) (A + B) \* D + E / (F + A \* D) + C

(d) A && B || C || !(E > F)

(e) !(A && !((B < C) || (C > D))) || (C < E)

Ans:

|  | postfix | prefix |
| --- | --- | --- |
| (a) | A -B+C-D+ | +-+-ABCD |
| (b) | AB-\*C+ | +\*A-BC |
| (c) | AB+D\*EFAD\*+/+C+ | ++\*+ABD/E+F\*ADC |
| (d) | AB&&C||EF>!|| | || ||&&ABC!>EF |
| (e) | ABC<CD> ||!&&!CE< || | ||!&&A!||<BC>CD<CE |