3) (10 pts) ALG (Queues)

Consider the circular array implementation of a queue named Q, implemented with the structure shown below.

```
struct queue {
    int *array;
    int num_elements;
    int front;
    int capacity;
};
```

Suppose the queue is created with a capacity of 5 and front and num_elements are initialzed to 0. Trace the status of the queue by showing the valid elements in the queue and the position of front after each of the operations shown below. Indicate front by making bold the element at the front of the queue.

1. enqueue(Q, 50);
2. enqueue(Q, 34);
3. enqueue(Q, 91);
4. x = dequeue(Q);
5. enqueue(Q, 23);
6. y = dequeue(Q);
7. enqueue(Q, y);
8. enqueue(Q, 15);
9. enqueue(Q, x);
10. x = dequeue(Q);

After stmt #1:

front

50				

After stmt #2:

front

50	34			

After stmt #3:

front

50	34	91		

After stmt #4:

front

	34	91		

After stmt #5:

front

	34	91	23	

After stmt #6: front

		91	23	

After stmt #7: front

		91	23	34

After stmt #8: front

15		**91**	23	34

After stmt #9: front

15	50	**91**	23	34

After stmt #10: front

15	50		**23**	34

Grading: 1 pt per array, must be perfectly correct to get the point.

Computer Science Foundation Exam

January 16, 2021

Section I B

DATA STRUCTURES

SOLUTION

Directions: You may either directly edit this document, or write out your answers in a .txt file, or scan your answers to .pdf and submit them in the COT 3960 Webcourses for the Assignment "Section I B". Please put your _name, UCFID and NID_ on the top left hand corner of each document you submit. Please aim to submit 1 document, but if it's necessary, you may submit 2. Clearly mark for which question your work is associated with. If you choose to edit this document, please remove this cover page from the file you submit and make sure your _name, UCFID and NID_ are on the top left hand corner of the next page (first page of your submission).

Question #	Max Pts	Category	Score
1	10	DSN	
2	5	ALG	
3	10	DSN	
TOTAL	25		

You must do all 3 problems in this section of the exam.

Problems will be graded based on the completeness of the solution steps and <u>not</u> graded based on the answer alone. Credit cannot be given unless all work is shown and is readable. Be complete, yet concise, and above all <u>be neat</u>. For each coding question, assume that all of the necessary includes (stdlib, stdio, math, string) for that particular question have been made.