
Sheet #2 (Queue)

1. Re-solve sheet 1 but for Circular Queue ADT.
2. Write a function that returns the last element in a queue. (Implementation level)
3. Write a function that returns a copy from the first element in a queue. (Implementation level)
4. Write a function to destroy a queue (Implementation level)
5. Write a function to copy a queue to another. (Implementation level)
6. Write a function to return the size of a queue (Implementation level)
7. Write a function that returns the last element in a queue. (User level)
8. Write a function that returns a copy from the first element in a queue. (User level)
9. Write a function to destroy a queue (User level)
10. Write a function to copy a queue to another. (User level)
11. Write a function to return the size of a queue (User level)
12. Use a stack structure to check the balance and ordering between various parentheses.
13. We (as a user for QueueADT) have two filled queues; the first queue holds section code while the other holds group code (where number of groups inside the section is maximum 10). Merge those numbers (section code*10+group code) in a newly created queue.
14. We (as a user for StackADT) have a stack holding group_ids. Each group_id consists of two parts section code and group code within his section. Number of groups inside the section is maximum 10. $\text{section_code} = \text{group_id} / 10$, $\text{group_code} = \text{group_id} \% 10$. Construct two stacks; one stack holds section codes while the other holds group codes.