

# CP2410 Practical 04 - Stacks, Queues, and Deques

1. (R-6.1) What values are returned during the following series of stack operations, if executed upon an initially empty stack?  
push(5), push(3), pop(), push(2), push(8), pop(), pop(), push(9), push(1), pop(), push(7), push(6), pop(), pop(), push(4), pop(), pop().
2. (R-6.2) Suppose an initially empty stack S has executed a total of 25 push operations, 12 top operations, and 10 pop operations, 3 of which raised Empty errors that were caught and ignored. What is the current size of S?
3. (R-6.3) Implement a function **transfer(S, T)** that transfers all elements from stack S onto stack T, so that the element that starts at the top of S is the first to be inserted onto T, and the element at the bottom of S ends up at the top of T. Use ArrayStack (ch06/array\_stack.py) to test your function.
4. (R-6.7) What values are returned during the following sequence of queue operations, if executed on an initially empty queue?  
enqueue(5), enqueue(3), dequeue(), enqueue(2), enqueue(8), dequeue(), dequeue(), enqueue(9), enqueue(1), dequeue(), enqueue(7), enqueue(6), dequeue(), dequeue(), enqueue(4), dequeue(), dequeue().
5. (R-6.8) Suppose an initially empty queue Q has executed a total of 32 enqueue operations. 15 dequeue operations were also executed, 5 of which raised Empty errors that were caught and ignored. What is the current size of Q?
6. (R-6.13) Suppose you have a deque D containing the numbers (1,2,3,4,5,6,7,8), in this order. Suppose further that you have an initially empty queue Q. Give a code fragment that uses only D and Q (and no other variables) and results in D storing the elements in the order (1,2,3,5,4,6,7,8).
7. (R-6.14) Repeat the previous problem using the deque D and an initially empty stack S.