1. Given an integer k and a queue of integers, reverse the order of the first k elements of the queue, leaving the other elements in the same relative order.
2. Given a queue of integers of even length, rearrange the elements by interleaving the first half of the queue with the second half of the queue.

1. Use stack
2. Use queue

3. Given a Queue consisting of first n natural numbers (in random order). check whether the given Queue elements can be arranged in increasing order in another Queue using a stack.

4. Below you find a list of functions that could appear as functions describing the running time of algorithms:

1. n^3/2
2. 8 ^logn
3. 2.3^n
4. 3.n^2
5. 3.2^n
6. 3.n^2+2.n^3
7. n^log7
8. Log n^n

Order the functions according to their asymptotic complexity. Start with the function having the smallest asymptotic complexity and move on to the function having the next largest one. That is, write them as a sequence f1, f2, . . . , f9 such that f1(n) = O(f2(n)), f2(n) = O(f3(n)), etc. Indicate as well functions f, g that are asymptotically equivalent, that is, where both f(n) = O(g(n)) and g(n) = O(f(n)) hold.

5. Prove or disprove the following statements:

a) $8n + n \cdot \log_2 n = O(n)$

b) $(n + a)\, b = \Omega(n\, b)$ for all real numbers a, b > 0

c) $n\, a = \Theta(n\, b)$ if and only if a = b for all real numbers a, b > 0