Exercises with Doubly Linked Lists

Exercise 1:

Write the following functions, assuming that they will be applied over a doubly linked list containing integers: Insert, Search and Delete. Consider implementations targeting sorted and unsorted doubly linked lists.

Exercise 2:

Suppose you have a doubly linked list containing a sequence of *n* integers sorted in ascending order. Write a program that create, from the given doubly linked list, two singly linked lists. The first singly linked list should contain the same sequence of *n* integers sorted in ascending order found in the doubly linked list. The second singly linked list should contain the same sequence of *n* integers found in the doubly linked list, but in descending order. The construction of the two singly linked lists must be made simultaneously and present a runtime cost of $O(n)$.

Exercise 3: ***from the book Introduction to Algorithms, Cormen T.H. et al.***

Explain how to implement doubly linked lists using only one pointer value x.np per item instead of the usual two (next and previous). Assume that all pointer values can be interpreted as k-bit integers, and define x.np to be x.np = x.next XOR x.previous, the k-bit “exclusive-or” of x.next and x.previous (the value NULL is represented by 0). Be sure to describe what information you need to access the head of the list. Show how to implement the Search, Insert, and Delete operations on such a list. Also, show how to reverse such a list in O(1) time.