3. Using stack ADT, implement a program to check for Balanced Brackets in an expression. The input will be given as an expression “string”
4. Write a program to check if a queue can be sorted into another queue using a stack (using push/pop, enqueue/dequeue operations only.)
5. Write a program to rearrange the elements of a given queue of integers of even length by interleaving the first half of the queue with the second half of the queue.
6. Write a program to check if a given linked list is a palindrome.
7. Write a program to reverse a linked list.
8. Write a program to calculate the minimal cost of connecting n ropes. You are given n ropes of different length, connect them into single rope with minimum cost. You can assume the cost of connecting two ropes is same as the sum of their lengths. Use heap for this problem.

   Input: [5, 4, 2, 8]

   Output: The minimum cost is 36

   [5, 4, 2, 8] –> First, connect ropes of lengths 4 and 2 that will cost 6.
   [5, 6, 8] –> Next, connect ropes of lengths 5 and 6 that will cost 11.
   [11, 8] –> Finally, connect the remaining two ropes that will cost 19.

   Therefore, the total cost for connecting all ropes is 6 + 11 + 19 = 36
9. calculate the frequency of all elements present in a limited range array. You are given an unsorted integer array whose elements lie in the range of 0 to n-1 where n is array size. Calculate how many times all elements appear in the given array/list. Use dictionary for this.
10. Implement a class for BST with the interfaces for adding elements , deleting elements, traversals(3 DFS),search for the minimum and maximum, creating a bst from a given array and size of the bst.

Note: Try to implement all these as part of a Class. Write function/methods for each specific task.