1. Given a queue of integers, rearrange the elements by interleaving the first half of the list with the second half of the list. For example, suppose a queue stores the following sequence of values: [11, 12, 13, 14, 15, 16, 17, 18, 19, 20]. Consider the two halves of this list: first half: [11, 12, 13, 14, 15] second half: [16, 17, 18, 19, 20]. These are combined in an alternating fashion to form a sequence of interleave pairs: the first values from each half (11 and 16), then the second values from each half (12 and 17), then the third values from each half (13 and 18), and so on. In each pair, the value from the first half appears before the value from the second half. Thus, after the call, the queue stores the following values: [11, 16, 12, 17, 13, 18, 14, 19, 15, 20].
2. Implement Enqueue, Dequeue and Display functions for Queue which is implemented by two stacks.
3. For a given *K* value (*K >* 0) reverse blocks of *K* nodes in a single linked list.

**Example:** Input: 1 2 3 4 5 6 7 8 9 10. Output for different *K* values:

For *K =* 2: 2 1 4 3 6 5 8 7 10 9

For *K =* 3: 3 2 1 6 5 4 9 8 7 10

For *K =* 4: 4 3 2 1 8 7 6 5 9 10

1. Write a program to Split a Circular doubly Linked List into two equal parts. If the number of nodes in the list are odd then make first list one node extra than second list.
2. Write a program to find the following in the binary tree without using recursion:
3. the number of full nodes ( nodes with two children)
4. number of half nodes (*n*odes with only one child)
5. the number of leaf nodes,
6. print all the ancestors of a node
7. Given a BST and two integers (minimum and maximum integers) as parameters, write a program to remove (prune) elements that are not within that range.
8. Given an array A[] consisting of 0’s, 1’s and 2’s, Write a program to sort this array *A*[] using Quick Sort.
9. Write a program for finding the *kth* smallest element in min-heap.
10. Write a program for finding the (i) maximum-weight spanning tree (ii) minimum weight spanning tree in a graph.
11. Write a program to implement Traveling Salesman Problem using Dynamic Programming..