**PSG College of Technology**

**Department of AMCS**

**II Sem M.Sc. CS Data Structures Lab**

**Important Linked Lists Programs**

1. Divide a singly linked list such that nodes with odd values from one sublist and nodes with even values form another sublist.
2. Given a singly linked list and two keys in it, swap nodes for two given keys. Nodes should be swapped by changing links. Swapping data of nodes may be expensive in many situations when data contains many fields. It may be assumed that all keys in the linked list are distinct.

**Examples:**

**Input :** 10->15->12->13->20->14, x = 12, y = 20

**Output:** 10->15->20->13->12->14

**Input :** 10->15->12->13->20->14, x = 10, y = 20

**Output:** 20->15->12->13->10->14

1. Given two circular lists sorted in increasing order, create and return a new singly linked list representing the intersection of the two lists.

**Example:**

**Input:**

First linked list: 1 -> 2 -> 3 -> 4 -> 6 -> 1

Second linked list: 2 -> 4-> 6-> 8 -> 2,

**Output:** **2->4->6.**

1. Exchange the first and last node of a doubly linked list.

Example:

Input list : 1 <-> 2 <-> 3 <-> 4 <-> 5 <-> 6

Output list : 6 <-> 2 <-> 3 <-> 4 <-> 5 <-> 1

1. Reverse a doubly linked list

Input list : 1 <-> 2 <-> 3 <-> 4 <-> 5 <-> 6.

Output list : 6 <-> 5 <-> 4 <-> 3 <-> 2 <-> 1.