

## Week 7 – Revision Exercise

### Task#1

- Delete the node from front, last and at any position and display the updated list.
- Given the head of a Doubly Linked List, reverse the list in-place so that the first node becomes the last, the second node becomes the second last, and so on. Return the new head of the reversed list.

**Note:** use the recursive approach to achieve this task.

### Task#2

- Sort the linked list using bubble sort.**
- Given two singly linked lists, the task is to insert nodes of the second list into the first list at alternate positions of the first list and leave the remaining nodes of the second list if it is longer.

**Input:**

**head1:** 1->2->3 , **head2:** 4->5->6->7->8

**Output:**

**head1:** 1->4->2->5->3->6 , **head2:** 7->8

**Task#3:** Consider the below maze, start from **(0, 0)** and target is to reach at **(2, 0)**. Write a backtracking code in c++ to achieve this. Note: **1** is open and **0** is blocked.

**Maze**

1 1 1 1

0 0 0 1

1 1 0 1

1 1 1 1

**Task#4:** Consider that you have a chessboard of size  $N \times N$ , You need to place  $N$  queens such that: No two queens share the same row, No two queens share the same column, No two queens share the same diagonal (both left and right diagonals).

A queen in chess can move:

- Vertically (up and down)
- Horizontally (left and right)
- Diagonally (both directions).

So, we must place them safely.

**Example:**  $N = 4$

- Q - -

- - - Q

Q - - -

- - Q -