1(a)

took adjacency Matrix, and took input in it, where there is a edge there adj[i][j] = 1. Lastly print the final Matrix.

1(b)

Took adjacency list of length of n(V), append on its column index to row index point as a edge & its value will be equal to 0. Because it a adjacency list. finally print the adjacency list.

2) BFS Traversal with the help of BFS Algorithm. its just plain BFS Traversal with source / start point as 1.

(3) it's pure DFS Traversal. following to the dfs algorithm. it will start traversing from vertex 1. & print the vertex when visited.

(2) for cycle detection. I used DFS algorithm. By the time of DFS algorithm ~~I store every adjacency node's parent node when not visited.~~ when it gets a already visited node, it will return true as there is a cycle. Then just need to check if there any cycle exist or not.

(5) BFS Traversal result the shortest path. By storing the parent vertex of each unvisited adjacent vertex. It will give the path result. Here we need to track back from destination to source to get the right path of the shorted distance from start.

(6) For Flood fill I used DFS algorithm. It will go deepest possible vertex by recursion. We have to count the '1' vertex and finally return it. We must avoid the restriction the question have to get the actual result.