Problem 1. Represent the graph below in Ruby using the adjacency list representation.

```
N = 8
G = Array.new
N.times{|i| G[i]=Array.new}
G[0]<<1<<2
G[1]<<0<<2
G[2]<<0<<1<<3<<4
G[3]<<2
G[4]<<2
G[5]<<7
G[6]<<7
G[7]<<5<<6
```

Problem 2. Do a DFS traversal of the graph starting from node 3. Write down the node numbers in the order they are visited.

A possible answer would be:

Start at 3. Mark 3 as visited. Then visit its neighbor 2. From 2, visit its neighbor 4. Come back to 2 since 4 has all neighbors visited. Visit 0. It has a neighbor 1 that is not yet visited.
Visit 1 from 0. 1 has all neighbors visited. Come back to 0. Now all neighbors of 0 are visited.
Come back to 2. Now 2 has all neighbors visited. Come back to 3. Since all neighbors of 3 are visited, the DFS traversal ends here. The sequence of node visits are: 3, 2, 4, 0, 1.

The following diagram illustrates this. The green arrows indicate the first visit of a node and the black arrows indicate the returns.