# CS69003: Computing Systems Lab I
# Autumn 2008

## Assignment 3

## Eulerian circuits in simple undirected graphs

## Due: August 22, 2008 (Friday)

In this exercise, you are required to implement *Fleury's algorithm* for computing Eulerian circuits in simple undirected graphs.

### Part I (10)

Represent an undirected graph by its *adjacency matrix*. For simplicity, assume that we deal with simple graphs only, that is, graphs without loops and multiple edges. Declare a suitable C data type for storing the adjacency matrix of a graph.

Read a graph from a text file. The first line of the file stores the number $n$ of vertices in the graph and the second line stores the number $e$ of edges in the graph. Number the vertices in the graph as $1, 2, \ldots, n$. In each of the lines 3 through $e + 2$ of the file, the information of an edge is stored as a pair of distinct integers representing the two vertices that the edge connects. You may assume that $n \leq 100$.

### Part II (30)

Write a C function that takes the adjacency matrix of a graph as input and returns the decision whether the graph is connected or not. Use a standard graph traversal algorithm to implement this function.

### Part III (10)

An *Eulerian circuit* or *Eulerian tour* in a graph is a path in the graph that traverses each edge of the graph once and only once and returns to the same vertex from which it started. Note that vertices may be repeated in an Eulerian circuit. A graph containing an Eulerian circuit is called an *Eulerian graph*.

It is known (Carl Hierholzer, 1873) that a graph is Eulerian if and only if it is connected and each vertex of it has even degree. Write a C function that, upon the input of the adjacency matrix of a graph, determines whether the graph is Eulerian.

### Part IV (50)

Write a C function to construct an Eulerian circuit in a graph. Let $G$ denote the graph input to the function. If $G$ is not Eulerian, your function should report failure. Otherwise, proceed as follows.

First, choose any arbitrary vertex $v_1$ in $G$ and start the Eulerian circuit from that vertex. Suppose that vertices $v_1, v_2, \ldots, v_i$ are computed in the Eulerian circuit (there may be repetitions of vertices in this list). For obtaining the next vertex $v_{i+1}$ in the circuit, look at all the (remaining) neighbors of $v_i$ in the graph. If there is only one neighbor of $v_i$, take that neighbor as $v_{i+1}$. In the case of multiple neighbors, choose $v_{i+1}$ to be a neighbor such that the removal of the edge $(v_i, v_{i+1})$ retains a connected graph. If $G$ is Eulerian, such a neighbor is guaranteed to exist. If several neighbors satisfy this criterion, choose any one of them arbitrarily. Delete the edge $(v_i, v_{i+1})$ from $G$. Moreover, if all edges adjacent to $v_i$ have been already deleted, delete the vertex $v_i$ from $G$. Repeat this march in the graph until an Eulerian circuit is completely constructed. Print the sequence of vertices in the Eulerian circuit in the order they are discovered.

---

You are required to submit a single C file with the name `<your_roll_no>-assgn3.c` solving all the parts of this assignment.