

## 8 Puzzle Solver

A very popular board puzzle is the 8-puzzle. It consists of 8 square blocks numbered 1 to 8, arranged on a  $3 \times 3$  board with one block empty (denoted by 0).

1	5	7
4	0	2
3	8	6

Any block on the left, right, top or bottom of the blank block can be shifted to the blank position. For example, shifting the block 4 to right results in the configuration,

1	5	7
0	4	2
3	8	6

The goal is to shift the blocks in such way so that we reach the final configuration in which the numbers are arranged consecutively as in the following figure.

0	1	2
3	4	5
6	7	8

In this exercise, your goal is to create a solver for the 8-puzzle, so that given an initial configuration of the blocks, it gives us the consecutive valid arrangements of the blocks which should be made in order to reach the final configuration. A convenient way to solve it, is to consider each configuration of the blocks as a node in the graph and edges between nodes which can be reached by one shift of any blocks. Then to start searching from the given configuration and figure out a way to reach the final configuration. The input is given as a  $3 \times 3$  integer matrix with the blank tile marked by 0. It should print "unsolvable" if there is no way to solve the puzzle from the given position. Note that the number of different configurations could be high (9!). To do it efficiently, apply some optimization techniques (representation of configurations, exploration of only necessary configurations etc.).

### Notes:

- Try to make the program as dynamic as possible.
- Please use only C programming language for the assignment.