#include <stdio.h>

#include <stdbool.h>

#define N 9

void printGrid(int grid[N][N]) {

for (int row = 0; row < N; row++) {

for (int col = 0; col < N; col++) {

if (col == 3 || col == 6)

printf(" | ");

printf("%d ", grid[row][col]);

}

if (row == 2 || row == 5) {

printf("\n");

for (int i = 0; i < N; i++)

printf("---");

}

printf("\n");

}

}

bool isValid(int grid[N][N], int row, int col, int num) {

for (int x = 0; x < N; x++) {

if (grid[row][x] == num)

return false;

if (grid[x][col] == num)

return false;

if (grid[row - row % 3 + x / 3][col - col % 3 + x % 3] == num)

return false;

}

return true;

}

bool solveSudoku(int grid[N][N]) {

int row = -1, col = -1;

bool isEmpty = true;

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++) {

if (grid[i][j] == 0) {

row = i;

col = j;

isEmpty = false;

break;

}

}

if (!isEmpty)

break;

}

if (isEmpty)

return true;

for (int num = 1; num <= N; num++) {

if (isValid(grid, row, col, num)) {

grid[row][col] = num;

if (solveSudoku(grid))

return true;

grid[row][col] = 0;

}

}

return false;

}

int main() {

int grid[N][N] = {

{5, 3, 0, 0, 7, 0, 0, 0, 0},

{6, 0, 0, 1, 9, 5, 0, 0, 0},

{0, 9, 8, 0, 0, 0, 0, 6, 0},

{8, 0, 0, 0, 6, 0, 0, 0, 3},

{4, 0, 0, 8, 0, 3, 0, 0, 1},

{7, 0, 0, 0, 2, 0, 0, 0, 6},

{0, 6, 0, 0, 0, 0, 2, 8, 0},

{0, 0, 0, 4, 1, 9, 0, 0, 5},

{0, 0, 0, 0, 8, 0, 0, 7, 9}

};

printf("Unsolved Sudoku:\n");

printGrid(grid);

if (solveSudoku(grid)) {

printf("\nSolved Sudoku:\n");

printGrid(grid);

} else {

printf("\nNo solution exists.\n");

}

return 0;

}