#include <iostream>

#include <string>

using namespace std;

const int N = 9; // Grid size

// Function to print the Sudoku board

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

cout << "\n 1 2 3 4 5 6 7 8 9\n";

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

if (row % 3 == 0)

cout << " +-------+-------+-------+\n";

cout << row + 1 << " |";

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

if (grid[row][col] == 0) cout << " ";

else cout << " " << grid[row][col];

if ((col + 1) % 3 == 0) cout << " |";

}

cout << "\n";

}

cout << " +-------+-------+-------+\n";

}

// Check if placing num is valid

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

// Row check

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

if (grid[row][x] == num) return false;

// Column check

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

if (grid[x][col] == num) return false;

// 3×3 box check

int startRow = row - row % 3, startCol = col - col % 3;

for (int i = 0; i < 3; i++)

for (int j = 0; j < 3; j++)

if (grid[i + startRow][j + startCol] == num)

return false;

return true;

}

// Backtracking Sudoku solver

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

int row = -1, col = -1;

bool emptyFound = false;

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

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

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

row = i;

col = j;

emptyFound = true;

}

}

}

if (!emptyFound) return true; // Solved

for (int num = 1; num <= 9; num++) {

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

grid[row][col] = num;

if (solveSudoku(grid)) return true;

grid[row][col] = 0; // backtrack

}

}

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}

};

cout << "Welcome to Sudoku!\n";

cout << "Commands:\n";

cout << " - Enter: row col number (e.g., 3 4 5)\n";

cout << " - Type 'solve' to auto-solve\n";

cout << " - Type 'exit' to quit\n";

while (true) {

printGrid(grid);

cout << "Enter command: ";

string cmd;

cin >> cmd;

if (cmd == "exit") {

cout << "Goodbye!\n";

break;

}

else if (cmd == "solve") {

if (solveSudoku(grid)) {

cout << "Puzzle solved!\n";

printGrid(grid);

} else {

cout << "No solution found!\n";

}

break;

}

else {

int row, col, num;

try {

row = stoi(cmd); // first value was already read into cmd

cin >> col >> num;

} catch (...) {

cout << "Invalid input!\n";

continue;

}

if (row < 1 || row > 9 || col < 1 || col > 9 || num < 1 || num > 9) {

cout << "Values must be between 1 and 9!\n";

continue;

}

if (grid[row-1][col-1] != 0) {

cout << "Cell already filled!\n";

continue;

}

if (!isSafe(grid, row-1, col-1, num)) {

cout << "Invalid move according to Sudoku rules!\n";

continue;

}

grid[row-1][col-1] = num;

}

}

return 0;

}