#include <iostream>

using namespace std;

const int SIZE = 9;

void displayBoard(int sudokuBoard[SIZE][SIZE])

{

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

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

{

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

{

if (j % 3 == 0)

{

cout << "| ";

}

cout << sudokuBoard[i][j] << ' ';

}

cout << "|\n";

if ((i + 1) % 3 == 0) {

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

}

}

}

bool isValidMove(int sudokuBoard[SIZE][SIZE], int row, int col, int num)

{

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

{

if (sudokuBoard[row][i] == num || sudokuBoard[i][col] == num)

{

return false;

}

}

int startRow = 3 \* (row / 3);

int startCol = 3 \* (col / 3);

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

{

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

{

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

{

return false;

}

}

}

return true;

}

bool isBoardFull(int sudokuBoard[SIZE][SIZE])

{

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

{

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

{

if (sudokuBoard[i][j] == 0)

{

return false;

}

}

}

return true;

}

void playSudoku()

{

int sudokuBoard[SIZE][SIZE] = {

{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";

displayBoard(sudokuBoard);

while (!isBoardFull(sudokuBoard))

{

int row, col, num;

cout << " Enter row (1-9), column (1-9), and number (1-9) to place (e.g., 3 5 7): ";

cin >> row >> col >> num;

--row;

--col;

if (row >= 0 && row < SIZE && col >= 0 && col < SIZE && num >= 1 && num <= 9)

{

if (sudokuBoard[row][col] == 0 && isValidMove(sudokuBoard, row, col, num))

{

sudokuBoard[row][col] = num;

displayBoard(sudokuBoard);

}

else

{

cout << "Invalid move. Try again.\n";

}

}

else

{

cout << "Invalid input. Please enter valid row, column, and number.\n";

}

}

cout << "Congratulations! You've completed the Suduk0 puzzle.\n";

}

int main()

{

playSudoku();

return 0;

}