#include <iostream>

using namespace std;

#define N 9

void print(int arr[N][N])

{

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

{

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

cout << arr[i][j] << " ";

cout << endl;

}

}

bool isSafe(int grid[N][N], int row,

int col, int num)

{

for (int x = 0; x <= 8; x++)

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

return false;

for (int x = 0; x <= 8; x++)

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

return false;

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;

}

bool solve(int grid[N][N], int row, int col)

{

if (row == N - 1 && col == N)

return true;

if (col == N) {

row++;

col = 0;

}

if (grid[row][col] > 0)

return solve(grid, row, col + 1);

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

{

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

{

grid[row][col] = num;

if (solve(grid, row, col + 1))

return true;

}

grid[row][col] = 0;

}

return false;

int main()

{

int grid[9][9] = { { 3, 0, 6, 5, 0, 8, 4, 0, 0 },

{ 5, 2, 0, 0, 0, 0, 0, 0, 0 },

{ 0, 8, 7, 0, 0, 0, 0, 3, 1 },

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

{ 9, 0, 0, 8, 6, 3, 0, 0, 5 },

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

{ 1, 3, 0, 0, 0, 0, 2, 5, 0 },

{ 0, 0, 0, 0, 0, 0, 0, 7, 4 },

{ 0, 0, 5, 2, 0, 6, 3, 0, 0 } };

if (solve(grid, 0, 0))

print(grid);

else

cout << "no solution exists " << endl;

return 0;

}