#include <stdio.h>

#define SIZE 9

//sudoku problem

int matrix[9][9] = {

{6,5,0,8,7,3,0,9,0},

{0,0,3,2,5,0,0,0,8},

{9,8,0,1,0,4,3,5,7},

{1,0,5,0,0,0,0,0,0},

{4,0,0,0,0,0,0,0,2},

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

{5,7,8,3,0,1,0,2,6},

{2,0,0,0,4,8,9,0,0},

{0,9,0,6,2,5,0,8,1}

};

//function to print sudoku

void print\_sudoku()

{

int i,j;

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

{

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

{

printf("%d\t",matrix[i][j]);

}

printf("\n\n");

}

}

int number\_unassigned(int \*row, int \*col)

{

int num\_unassign = 0;

int i,j;

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

{

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

{

//cell is not assigned yet

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

{

//altering the values of row and colums trying to solve

\*row = i;

\*col = j;

num\_unassign = 1;

return num\_unassign;

}

}

}

return num\_unassign;

}

int is\_safe(int n, int r, int c)

{

int i,j;

//checking in row

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

{

if(matrix[r][i] == n)

return 0;

}

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

{

if(matrix[i][c] == n)

return 0;

}

int row\_start = (r/3)\*3;

int col\_start = (c/3)\*3;

for(i=row\_start;i<row\_start+3;i++)

{

for(j=col\_start;j<col\_start+3;j++)

{

if(matrix[i][j]==n)

return 0;

}

}

return 1;

}

//function to solve sudoku

//using backtracking using c algo

int solve\_sudoku()

{

int row;

int col;

if(number\_unassigned(&row, &col) == 0)

return 1;

int n,i;

for(i=1;i<=SIZE;i++)

{

if(is\_safe(i, row, col))

{

matrix[row][col] = i;

//backtracking

if(solve\_sudoku())

return 1;

matrix[row][col]=0;

}

}

return 0;

}

int main()

{

if (solve\_sudoku())

print\_sudoku();

else

printf("No solution\n");

return 0;

}