Here's the Code:

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#define RESET 0

int sudokuSolver();

int findEmptyCell();

int isValid();

void printGrid();

void inputGrid();

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

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

{0, 7, 0, 0, 1, 0, 5, 0, 0},

{4, 0, 0, 0, 0, 5, 3, 0, 0},

{0, 1, 0, 0, 7, 0, 0, 0, 6},

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

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

{0, 0, 4, 0, 0, 0, 0, 3, 0},

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

};

int row,col;

//this variable was used just to keep track of number of recursive calls

long int totalNumOfCalls=0;

void main(){

int i,j,solution=0;

char ch;

clrscr();

printf("You can change the puzzle before running the program \nby changing the values in the \"grid\" array\n\n");

printf("The Entered Sudoku puzzle is: \n");

printGrid();

printf("Press 'c' to confirm and solve, or 'e' to exit: ");

ch=getch();

if(ch=='e')

exit(0);

else if(ch=='c'){

clrscr();

solution=sudokuSolver();

if(solution){

printf("\nThe Solved Sudoku is: \n\n");

printGrid();

}

else

printf("\n No Possible Solution!!\n\n");

getch();

}

}

int findEmptyCell(){

int i,j;

for(i=row;i<=8;i++)

for(j=0;j<=8;j++){

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

{

row=i;col=j;

return 1;

}

}

return 0;

}

int isValid(int cellRow, int cellCol, int num){

int i,j,trow,tcol,trow1,tcol1;

int rowStart = (cellRow/3) \* 3;

int colStart = (cellCol/3) \* 3;

// to check the presence of number in row and column

for(j=0;j<=8;j++){

if(grid[cellRow][j]==num)

return 0;

if(grid[j][cellCol]==num)

return 0;

}

// to check the presence of number in 3X3 box

for(i=rowStart;i<=rowStart+2;i++)

for(j=colStart;j<=colStart+2;j++)

if(grid[i][j]==num)return 0;

return 1;

}

int sudokuSolver(){

int digit;

int prevRow,prevCol; // for backtracking

totalNumOfCalls++;

if(!findEmptyCell())

return 1;

for(digit=1;digit<=9;digit++){

if(isValid(row,col,digit)){

grid[row][col]=digit;

prevRow=row;prevCol=col;

if(sudokuSolver())

return 1;

//while backtracking assigning previous values to row and col

row=prevRow;col=prevCol;

grid[row][col]=RESET;

}

}

return 0;

}

void printGrid(){

int i,j;

printf("\t-------------------------\n");

for(i=0;i<9;i++){

printf("\t");

for(j=0;j<9;j++){

if(j==0)

printf("| ");

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

printf(". ");

else

printf("%d ",grid[i][j]);

if((j+1)%3==0 )

printf("| ");

}

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

printf("\n\t-------------------------");

printf("\n");

}

}