//Sudokusolve.java

//https://www.youtube.com/watch?v=G\_UYXzGuqvM

//

public class Sudokusolve {

static int[][] arr =

{{0,0,0,0,0,6,7,0,9},

{4,5,6,7,8,9,1,2,3},

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

{2,3,1,5,6,4,0,0,7},

{0,0,4,0,9,7,2,0,1},

{8,9,0,2,3,0,5,6,4},

{3,1,2,6,0,0,0,0,0},//rechts am Rand: 7,8.

{6,4,5,0,7,0,3,0,2},//rechts: 3,1,2.

{9,0,0,0,0,0,0,4,5}};

public static void display()

{

 for(int y=0; y<9; y++)

 {

 for(int x=0; x<9; x++)

 System.out.print(arr[y][x] + " ");

 System.out.println();

 }//for

}

static boolean possible(int y, int x, int n)

{

 //Die Zeile testen:

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

 {

 if(arr[y][i]==n)

 return false;

 }//for

 //Die Spalte testen:

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

 {

 if(arr[i][x]==n)

 return false;

 }//for

 //Das kleine 3x3-Quadrat testen:

 int x0 = (x/3) \* 3;

 int y0 = (y/3) \* 3;

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

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

 if(arr[y0+i][x0+j]==n)

 return false;

 return true;

}//Funktion possible

public static void solve()

{

 for(int y=0; y<9; y++)

 for(int x=0; x<9; x++)

 if(arr[y][x]==0)

 {

 for(int n = 1; n<10; n++)

 {

 if(possible(y, x, n))

 {

 arr[y][x]=n;

 solve();//rekursiv

 arr[y][x]=0;//Backtracking

 }//if

 }//for

 return;//

 }//if

 //Anzeigen:

 display();

}//rekursive Funktion solve

public static void main(String[] args)

{

 System.out.println("Der Computer versucht, Sudoku zu spielen:");

 System.out.println();

 display();

 System.out.println();

 System.out.println("Und das bzw. die Ergebnis(se):");

 solve();

 System.out.println();

}

}//class