

## sudoku

Malo je poznato da je čuveni švicarski matematičar i fizičar Leonhard Euler tvorac i jedne zanimljive igre poznate pod imenom **Latin square**. **Latin square** je matrica  $n * n$  koja je popunjena sa  $n$  različitih simbola tako da se svaki simbol pojavljuje tačno jedanput u svakom redu i svakoj koloni. Npr.

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix} \quad \begin{bmatrix} a & b & d & c \\ b & c & a & d \\ c & d & b & a \\ d & a & c & b \end{bmatrix}$$

1979. je Howard Garns Latinskom kvadratu dodao 'treću' dimenziju, regionalnu restrikciju, i tako je nastala nova igra **SuDoku**. Originalni **SuDoku** je matrica  $9 * 9$  koja je podjeljena na 9 kvadrata  $3 * 3$  a koriste se znamenke od 1 do 9. Pravilo pojavljivanja različitih simbola/znamenki u svakom redu i svakoj koloni vrijedi i za svaki dodatni kvadrat  $3 * 3$ . Bertram Felgenhauer je 2005. 'izračunao' da postoji tačno 6,670,903,752,021,072,936,960 različitih **SuDoku** matrica. Taj broj je jednak  $9! * 72^2 * 27 * 27,704,267,971$ , gdje je zadnji faktor prost broj. Ed Russel i Frazer Jarvis su 'izračunali', da ako se isključe simetrične matrice ostaje 5,472,730,538 različitih rješenja. Budući nema smisla tražiti neko od tih rješenja, Garns je uveo i dodatno pravilo da se kreće od parcijalno popunjene matrice. To znači da se u početnoj matrici popune neke pozicije ali tako da postoji samo jedna 'puna' **SuDoku** matrica nastala od zadane početne pozicije. Npr.

iz									slijedi								
5	3			7					5	3	4	6	7	8	9	1	2
6			1	9	5				6	7	2	1	9	5	3	4	8
	9	8						6	1	9	8	3	4	2	5	6	7
8				6				3	8	5	9	7	6	1	4	2	3
4			8		3			1	4	2	6	8	5	3	7	9	1
7				2				6	7	1	3	9	2	4	8	5	6
	6					2	8		9	6	1	5	3	7	2	8	4
			4	1	9			5	2	8	7	4	1	9	6	3	5
				8			7	9	3	4	5	2	8	6	1	7	9

Mogući redoslijed rješavanja bi bio:

- i) budući se 5 nalazi u prvom i drugom redu te u devetoj koloni, jedino mjesto u gornjem desnom podkvadratu gdje se može nalaziti 5 je **osjenčeno**,
  - ii) u prvoj koloni nedostaje 9, a buduće se 9 već nalazi u trećem, osmom i devetom redu, jedino preostalo mjesto u prvoj koloni je u sedmom redu,
- i tako sve dok se ne popuni cijela matrica.

Vaš zadatak je da napišete program koji će za zadanu parcijalno popunjenu matricu naći i ispisati pripadnu SuDoku matricu. Za svaku zadanu parcijalno popunjenu matricu postoji tačno jedna pripadna SuDoku matrica.

## **sudoku**

### **Ulazni podaci (ASCII datoteka 'sudoku.in')**

Svaki red parcijalno popunjene matrice se nalazi u zasebnom redu. Znamenke su 'razdvojene' jednim praznim mjestom. Na poziciji znamenke koja nedostaje nalazi se 0 (nula).

### **Izlazni podaci (ASCII datoteka 'sudoku.out')**

Svaki red SuDoku matrice ispisati u zaseban red. Znamenke treba da budu razdvojene jednim praznim mjestom.

### **Primjer**

#### **sudoku.in**

```
0 8 9 7 4 1 2 5 3
7 1 4 3 2 5 8 6 9
5 2 0 8 6 9 1 7 4
2 7 1 9 3 4 6 8 5
8 9 5 2 0 6 3 4 1
4 3 6 1 5 8 7 9 2
3 4 2 5 8 7 0 1 6
1 5 7 6 9 2 4 3 8
9 6 8 4 1 3 5 2 0
```

#### **sudoku.out**

```
6 8 9 7 4 1 2 5 3
7 1 4 3 2 5 8 6 9
5 2 3 8 6 9 1 7 4
2 7 1 9 3 4 6 8 5
8 9 5 2 7 6 3 4 1
4 3 6 1 5 8 7 9 2
3 4 2 5 8 7 9 1 6
1 5 7 6 9 2 4 3 8
9 6 8 4 1 3 5 2 7
```